Spike-Based Computational Primitives for Vision-Based Scene Understanding

Dissertation
zur
Erlangung der naturwissenschaftlichen Doktorwürde

Dr. sc. nat.

vorgelegt der
Mathematisch-naturwissenschaftlichen Fakultät
der
Universität Zürich
von

Moritz Benjamin Milde

aus
Deutschland

Promotionskommission

Prof. Dr. Giacomo Indiveri
Dr. Matthew Cook
Prof. Dr. Daniel Kiper

Zürich, 2019

To family and friends

Disclaimer

I hereby declare, that the work in this thesis is that of the candidate alone, except where indicated in the text, and as described below. Every part of the project was shaped by discussions with Giacomo Indiveri, Matthew Cook and Elisabetta Chicca. Part I, including Chapter 1, is entirely original and represents an overview of relevant biological, theoretical and technological foundations that later chapters are based on. The work described here was conducted by others, as indicated by the corresponding references. Chapter 2 is an adapted version of the work that originally appeared in Milde, Blum, *et al.* (2017) and Milde, Dietmüller, *et al.* (2017). The work reported in Chapter 2 started as a project during the 2017 Capo Caccia Neuromorphic Engineering Workshop. Chapter 3 is an adapted version of the work that originally appeared in Milde, Bertrand, Ramachandran, *et al.* (2018). As a Master's project, Thorben Schoepe conducted closed-loop experiments using the spiking Elementary Motion Detector reported in the postamble of Chapter 3. Chapter 4 is entirely original. Many ideas it presents were discussed with Matthew Cook, Giacomo Indiveri, Gagan Narula, Travis Monk and Gregory Cohen. Chapter 5 is entirely original. Chapter 6 is entirely original. Some ideas were discussed with Matthew Cook and Elisabetta Chicca. Chapter 7 is entirely original. The use of "we" in this thesis refers to me, the candidate, but also to the input received from the above-mentioned people.

Acknowledgements

I would like to thank Matthew Cook for his guidance, his support and all the scientific and non-scientific discussions we had throughout the writing of this thesis. I am thankful to have him as an advisor and friend. I am grateful to Giacomo Indiveri who gave me the freedom to explore and always challenged my ideas, thus pushing me forward to exceed my limits. I am thankful to Elisabetta Chicca and Olivier Bertrand who inspired me in the early days of my scientific adventures and paved my way to become the scientist I am. Thanks to Gagan Narula for his genuine interest in my work, the beneficial discussions and his considerations on certain aspects of my model. I would like to thank Germain Haessig, Stan Kerstjens and Adrian Whatley for helpful comments on the manuscript.

Thanks to Richard (Riiihart) George for distracting me with side-side projects that kept me motivated throughout the years. I would like to further thank my friends Nils, Jubu, Nico, Benno, Saray, Gaby, Enea, Michi, Lenka, Lisanne, Alex, Fabian, Philip and finally the Spassgang who made my non-scientific life an absolute blast. I am grateful to Julia who has shared with me the highest and who supported me in the lowest moments of this work and lifted me over and over again. Thanks to my family who raised me the way they did and always supported me.

Last but not least I want to thank Kafischnaps without which this thesis would not have been possible.

Zusammenfassung

Nichts was wir Wahrnehmen ist statisch. Während die Zeit unaufhaltsam vorwärts schreitet detektieren unsere Sinnesorgane kontinuierlich einfache Formen in unserer Umgebung und unser Gehirn transformiert diese zu abstrakten Gedanken, Konzepten und Erinnerungen. Die zugrundeliegende Kontinuität der Zeit könnte der Schlüssel sein der es uns erlaubt verschiedene Wahrnehmungen auf einer Vielzahl von Zeitskalen miteinander zu assoziieren und könnte so als bindendes Glied agieren, dass uns erlaubt eine kohärente und stabile Vorstellung von Realität zu erschaffen. Im Gegensatz zu traditionellen Computern, sind die Schaltkreise im Gehirn durch eine extreme Parralelität und eine asynchrone Informationsverarbeitung charakterisiert, in welcher Berechnung und Erinnerung ko-loakilsiert sind. Die biologischen Schaltkreise reagieren in der Regel auf Änderungen in der sensorischen Wahrnehmungen und kommunizieren diese als Aktionspotentiale zwischen Nervenzellen mit einer hohen zeitlichen Präzision.

In den meisten Modellen, die genutzt werden um die Informationsverarbeitung des Gehirns nachzuempfinden, wird Information jedoch hauptsächlich in dem gemittelten Aktivitaetslevel kommuniziert und die präzise Zeit wann ein Aktionspotential generiert wird, wird in der Regel nicht als informativ betrachtet. Daraus lässt sich schliessen das unser aktuelles Verständnis von der Informationsverarbeitung in neuralen Netzwerken, unabhängig ob biologisch oder künstlich, primär ein raten Verständnis ist. Damit wir ein tieferes Verständnis der Vorteile für die Repräsentation von Informationen des zeitlichen Codes erlangen, haben wir neuronale Modelle entwickelt, welche Information aus der relativen Zeit von Aktionspotentialen extrahieren und diese in der zetilichen Struktur ihrer Antworten erhalten. Um unsere Modelle im Kontext von Echteit und anwendungsnahen Anwendungen zu validieren haben wir Neuromorphe Sensor-Prozessor Systeme benutzt, welche in der Lage sind, ähnlich wie das Gehirn, Zeit sich selbst repräsentieren zu lassen.

Um die Rolle von zeitlich präzisen Aktionspotenitalen zu studieren und welchen Effekt diese auf neuronale Informationsverarbeitung haben schlagen wir ein Spikendes Neuronales Netwerk vor welches an den rekurrenten Verbindungsmustern des Kortex inspiriert ist. Die Aufgabe dieses Netzwerk ist es räumlich-zeitliche Muster zu lernen und vorherzusagen. Das Netzwerk benutzt dazu ausschliesslich die räumlich-zeitliche Korrelation von Aktionspotentialen um rezeptive und projektierende Felder zu lernen. Die rekurrenten synaptischen Verbindungen sind mit lokalen Lernregeln ausgestattet welche das Netzwerk dazu ermöglichen Zeit inhärent zu repräsentieren. Das rekurrente Netwerk muss dazu in der Lage sein (i) stabile Berechnungen zu vollführen unabhängig von sensorischem und prozossorischem Rauschen und (ii) zeitliche Korrelation basierend auf einem zeitlichen Code zu berechnen. Beide Bedingungen haben wir untersucht bevor wir unser finales Modell konstruiert haben.

Mit diesen Untersuchungen hoffen wir ein wenig mehr Licht auf die zugrundeliegende Natur der Zeit in neuronaler Informationsverarbeitung zu werfen und wie Zeit als bindendes Glied agieren kann um räumlich-zeitliche Ativität zu lernen und vorherzusagen.

Abstract

Nothing we experience is static. As time progresses, our senses continuously capture the rich features of our environment which our brain transforms into abstract thoughts and memories. The continuous nature of time leads to causal relations between different stimuli across spatio-temporal scales and thus might be the key to construct a coherent and stable impression of reality. Unlike conventional computers, the computing structures of the brain (neurons and neural networks) are characterised by a massively parallel and asynchronous processing scheme with co-localised memory and computation. Neurons typically respond to changes in their sensory inputs and communicate them as action potentials, also called spikes, between processing elements, with high temporal precision. In our current understanding of information processing in both artificial and biological neural networks, streams of spikes are commonly reduced to instantaneous firing rates rather than considering the precise time of individual spikes. Thus, a potential source of information is irreversibly lost.

To further assess the computational benefits of precise temporal coding, we developed neurally-inspired models that use relative spike times to represent and process information. We validate these models in real-time and real-world scenarios by mapping them onto mixed-signal, sub-threshold neuromorphic sensory-processing systems, in which time represents itself.

We propose a Spiking Neural Network (SNN) that learns and predicts spatio-temporal patterns, whose highly recurrent structure is inspired by the neocortex. The proposed SNN exploits precise spike timing to form receptive fields and projective fields, using exclusively local, unsupervised learning rules. This SNN needs to (i) perform stable computation in the presence of sensory and processing noise and (ii) compute and represent temporal correlations using a temporal encoding, both of which we explored prior to constructing our final model. Our experiments expand on the nature of time in neural computation and how time can serve as a natural binding entity for learning and predicting spatio-temporal activity.

Contents

List of Acronyms

ADP Activity Dependent Plasticity 100
AER Address Event Representation 14
ANN Artificial Neural Network 5
ASE Adaptive Synaptic Efficacy 65
ATIS Asynchronous Time-based Image Sensor 29
aVLSI analogue Very Large Scale Integration 65
COMANV Center Of Mass Average Nearness Vector 83
CMOS complementary metal-oxide-semiconductor 17
CPU Central Processing Unit 6
DPI Differential Pair Integrator 65
DNF Dynamic Neural Field 37
DVS Dynamic Vision Sensor 13
DYNAP Dynamic Neuromorphic Asynchronuous Processor xvii
eDVS embedded Dynamic Vision Sensor 32
EMD Elementary Motion Detector 54
EMU Elementary Motion Unit 66
ESN echo-state network 101
EMG electromyography 5
EPSC Excitatory Post-Synaptic Current 69
fMRS functional Magnetic Resonance Spectroscopy 5
FPGA Field Programmable Gate Array 14
GPU Graphics Processing Unit 16
IPSC Inhibitory Post-Synaptic Current 99
ISI interspike interval 67
LIF leaky integrate-and-fire 65
LGMD Lobula Giant Movement Detector 54
LSTM Long Short Term Memory 22
MEA multi-electrode array 5
OCTA Online Clustering of Temporal Activity 100
OCTAPUS Online Clustering of Temporal Activity for Predictive Understanding of Scenes 143
PSC post-synaptic current 107
RC resistor–capacitor 4

ReLU Rectified Linear Unit 5
ROLLS Reconfigurable On-Line Learning System 18
ROS Robot Operating System 187
SNN Spiking Neural Network ix
SDRAM Synchronous Dynamic Random Access Memory 16
sEMD spiking Elementary Motion Detector 65
STDP Spike-Time Dependent Plasticity 24
WTA winner-takes-all 37

List of Figures

List of Tables

Objectives of this work: Visual scene understanding is a crucial requirement for an autonomously behaving system to successfully interact with the environment. In conventional, frame-based approaches time in computation is dictated by clocks (Krizhevsky *et al.,* 2012; Girshick *et al.,* 2014; Girshick, 2015; LeCun *et al.,* 2015; Long *et al.,* 2015; Ren *et al.,* 2015; He *et al.,* 2016). In unconventional, event-based approaches time in computation represents itself (Kramer *et al.,* 1995; Indiveri & Verschure, 1997; Kramer *et al.,* 1997; Maris & Mahowald, 1997; Etienne-Cummings *et al.,* 1999).

In neural network approaches to scene understanding information can be transmitted using a *rate* and/or a *temporal code* (Dayan & Abbott, 2001). There is evidence that suggests that the precise timing of spikes carries task-relevant information and in some cases is required for stable computation (Mainen & Sejnowski, 1995; Thorpe, Delorme, *et al.,* 2001; Wolfe *et al.,* 2008; Baudot *et al.,* 2013; Deneve & Machens, 2016; Goel & Buonomano, 2016). However, we still lack a complete understanding of how to take advantage of temporal coding in neural networks performing visual scene understanding.

To advance our understanding of the role of precise spike timing and its implication for neural computation based on a temporal code further we develop in this work neurally-implemented models to extract information from the relative timing of spikes. To enable down-stream processing stages also to operate using a temporal code the models preserve this temporal information in their outputs. To study the role of precise timing in neural computation, we propose a recurrent Spiking Neural Network (SNN), inspired by the highly recurrent circuit motifs of the neocortex, to learn and predict spatio-temporally correlated activity. Most of the synaptic projections within the recurrent SNN are subject to local, event-driven, unsupervised learning rules. We will demonstrate in detail how synaptic projections can incorporate spatio-temporally correlated features, temporal sequences and provide the necessary balance to enable stable processing. We thus hope to advance our understanding of temporal coding in SNNs in the context of visual scene understanding when the precise timing of spikes is considered to be informative.

PART I

The octopus on the watermelon

"The nature of time is perhaps the greatest remaining mystery."

– Carlo Rovelli, *The order of time*

In the absence of change, we will eventually stop perceiving the world around us. Having said that, it is possible to believe that a static image taken with a conventional imaging sensor emulates our percept of reality. However, there are fundamental differences between how the brain perceives and process information of the world, compared to conventional imaging sensors and computer vision processing systems. In the mid-1970s Stevens and colleagues conducted an experiment which had a fundamental impact on how we need to think about visual processing and perception (Stevens *et al.,* 1976). Among other conditions, Stevens tested visual perception under total paralysis using the depolarising blocking agent succinylcholine. The participant[1] was asked to perform voluntary eye movements, which due to the total paralysis could not be successfully executed. The participant reported that even under great effort trying to move his eyes, the world remained stationary. However, "[...] image fading became a real problem when the head was not moved." (Stevens *et al.,* 1976). The participant was certainly only able to visually perceive the environment "[...] due to inadvertant movements associated with artificial respiration." (Stevens *et al.,* 1976). This experiment shows the most convincing argument why conventional cameras capture reality quite differently from our visual apparatus. When we look for example at a completely stationary scene, like a forest on a windless day, we constantly perform microsaccadic eye movements, hence our eyes do not remain stationary. This has the effect that the photoreceptors are continuously subject to changes in the exposed luminance. If we start navigating in this scene, even more changes are induced and detected.

The inherent property of the retina and its down-stream processing architecture to respond to a change at the time of the change, make use of an asynchronous and parallel processing scheme (Squire *et al.,* 2012). This asynchronous information processing scheme allows the system to naturally represent time without the need of an explicit clock. Information is sensed and processed in a time-continuous manner. This is fundamentally different from conventional imaging sensors and computer vision processing systems, which sense and process at

[1] Due to missing ethical permission to perform the experiment with human participants it was Stevens himself who served as a subject.

a given (fixed) frequency (e.g. capturing visual inputs with a fixed frame-rate and a synchronous sensory read-out) (Yadid-Pecht & Etienne-Cummings, 2004). Biological sensory-processing systems, however, are time-continuous and incorporate stabilisation processes at both early and late processing stages (Tian & Lynch, 1996; Ilg, 1997; Krauzlis, 2005). Thanks to these stabilisation processes the world appears stable at any given moment in time (Robertson *et al.,* 2016) despite our sensory input needing to be continuously subject to changes (Stevens *et al.,* 1976; Ilg, 1997).

The role of time: In traditional computer programs, time is abstracted away from the computation, as any algorithm is defined by an explicit input-output function. For each input all desired outputs are computed as fast as possible depending on the system power budget, the processor's clock frequency and the complexity of the algorithm (see Section 1.7 in Patterson & Hennessy, 2014). Whether the output is generated faster or slower than real-time does not matter primarily for the computation. So we might think that analysing a static image to generate a desired output, e.g. the presence of a particular object, is sufficient to approximate an ongoing biological information processing of the very same scene. However, in biological and simulated neural networks subject to time-continuous inputs the point in time when an input is received or when internal activity is generated carries information (e.g. Thorpe, 1990; Mainen & Sejnowski, 1995, & see Chapter 1), and the time-continuous nature of computation provides the potential advantage of using time as a global orchestrating principle for low- and high-level processing (Zenke *et al.,* 2015; Goel & Buonomano, 2016). This is because generated behaviour, be it saccadic eye movements or navigating through an environment, induces changes at the sensory level which are causally linked through time. This inherent representation of time contribute to the perception of a coherent representation of visual scenes in humans (Robertson *et al.,* 2016), which constitutes the hallmark of artificial visual scene understanding (Arel *et al.,* 2010; LeCun *et al.,* 2015).
Let us assume that a given agent generates some behaviour, such as moving its visual apparatus, and assume further that its visual apparatus is able to perceive temporal contrast changes in a time-continuous manner. Then, the movement induced changes are temporally and spatially bounded by the coherent motion of objects. The activity sensed in the agent's visual apparatus can be associated

with different spatial scales using the temporal information present in the perceived sensory input. In a simplified scenario, changes which occur together in a spatio-temporal neighbourhood are likely to be induced by a common object, especially in the context of sub-millisecond temporal precision. Hence, the agent can associate not only spatio-temporal patterns in its low-level sensory input but also higher level feature-temporal patterns, if information remains precisely timed with constant system delay and so relative time differences are preserved throughout processing.

Capturing time in computation: A logical next question is what computational substrate would allow the agent to preserve information about timing in its processing stages and use time as a binding entity. In both electronic and biological neural circuits, time is most simply represented by the exponential amplification and decay (it is parametrised by a single time constant) of some state. More complex modules can be built by composing simple stateful units, given a means for these units to communicate (Doyle & Csete, 2011).
Not only is it the case that the bilipid membranes of neurons, which have the ability to separate a charge and accumulate charges over time, act as a capacitor, but the membrane is also penetrated by a myriad of ion channels with different temporal dynamics (Hodgkin & Huxley, 1952; Soldado-Magraner *et al.,* 2017), which act as voltage-dependent conductances and provide neurons with a multitude of time constants. The membrane of neurons can thus be described by a resistor–capacitor (RC) circuit (Hodgkin & Huxley, 1952; Squire *et al.,* 2012). Moreover, spiking neurons generate action potentials, i.e. spikes, which are initialised at the distal end of the axon initial segment (Coombs *et al.,* 1957; Kole *et al.,* 2008) less than a millisecond after its membrane potential crosses its threshold potential[2] (Dayan & Abbott, 2001; Kole *et al.,* 2008).

Limitations of computation: Despite these rather striking observations on the properties of biological spiking neurons, our theoretical understanding of compu-

[2]Such an explicit threshold, however, only exists in simplified neuron models, such as leaky integrate and fire neurons. In the conductance-based model of Hodgkin & Huxley (1952) the generation of an action potential arises from the interplay of channel dynamics rather than being explicitly specified to occur at a particular threshold.

tation in neural networks is governed mostly by networks in which information between neurons is encoded in the *output rate* of a given neuron, rather than in the *precise time* when a given neuron elicits an action potential or when a given neuron receives an incoming spike. The reason for this is twofold:
(I) In neuroscience, data obtained using measuring techniques, such as electromyography (EMG) (Marey, 1890), multi-electrode arrays (MEAs) (Thomas Jr *et al.,* 1972) and functional Magnetic Resonance Spectroscopy (fMRS) (Ogawa *et al.,* 1990), to record muscle activity and neuronal population activity based on local field potentials and large-scale metabolic processes, seem to better correlate with models encoding information in the firing rate of neurons due to the spatial-temporal resolution trade-off of these techniques (Sussillo *et al.,* 2015; Remington *et al.,* 2018). Also, data obtained using recording techniques with high temporal resolution, such as electrophysiology (Adrian, 1928) or patch clamp (Hamill *et al.,* 1981), suggest that single neurons encode information primarily in their output rate (Hubel & Wiesel, 1962; Shadlen & Newsome, 1998; Dayan & Abbott, 2001).
(II) In machine learning and Artificial Neural Networks (ANNs) the techniques commonly used to train ANNs, such as error-backpropagation (Rumelhart *et al.,* 1986) in combination with an optimisation procedure such as gradient descent (Cauchy, 1847; Robbins & Monro, 1951), require differentiating an error as produces by the network with respect to the weights. This is undoubtedly only possible if the neuron's activity is represented as a firing rate by a continuous, differentiable, non-linear function such as a step, sigmoidal or a Rectified Linear Unit (ReLU) transfer function (Nair & Hinton, 2010).

Alternative computation: What limits our understanding of neural computation is that we have little insight into encoding information in the timing of the spikes compared to encoding information in the output rate. This is because we lack a complete understanding of how spike timing could affect ongoing computation and processing. We are not first ones to study the effect and advantages of encoding information in the timing of spikes (e.g. Thorpe, 1990; Maass, 1997; Thorpe, Delorme, *et al.,* 2001; Bohte *et al.,* 2002; Gütig & Sompolinsky, 2006, & see Chapter 1) and we certainly have no complete answer to what the potential of SNNs is. But in order to leverage the full potential of these networks, we need to investigate the temporal aspect of neural computation further.

To study the temporal aspect of neural computation, we can not rely on the underlying logic of sequential processing schemes of Central Processing Units (CPUs)[3] and design algorithms accordingly, in which time is dictated by the processor's clock frequency. We instead need a system which features independent and parallel computational units that, as stated earlier, can reflect properties of biological neurons to naturally represent time in ongoing processing. When each component is independently operating, and computation is fully parallelised, it seems natural to not rely on a global clock which determines when to perform a specific computation, but instead, let activity, be it incoming or internal activity, dictate when to perform computation (Goel & Buonomano, 2016).
An alternative technology that offers such an event-driven, distributed and fully parallel processing scheme is that of neuromorphic computing. Neuromorphic sensory-processing systems have been developed for almost three decades (Mead, 1989; Liu, Kramer, *et al.,* 2002; Liu, Delbruck, *et al.,* 2014; Thakur *et al.,* 2018). Neuromorphic sensory-processing systems rely on sensing changes and transmitting spikes with sub-millisecond precision, in which each single component is independent and asynchronous. They provide both asynchronous and fully-parallel sensing (see Section 1.1) and processing (see Section 1.2). Due to the underlying asynchronous, data-driven logic, these systems naturally preserve temporal information and are time-continuous in their operation. In these systems "time represents itself" which precisely describes the fundamental difference to conventional synchronous sensory-processing systems.

A glimpse of visual scene understanding: The sensory-processing framework of neuromorphic systems enables these systems to naturally represent time. Let us focus now on what we intend these systems to do: understanding their visual environment or scene. Visual scene understanding is traditionally defined in the context of ANNs by analysing a static image to indicate the presence of a particular sub-class of objects, their location and ultimately assign each pixel to a high-level semantic category (Krizhevsky *et al.,* 2012; Girshick *et al.,* 2014; Girshick, 2015; Long *et al.,* 2015; Ren *et al.,* 2015; He *et al.,* 2016, & see Sec-

[3] This general architecture is referred to as von Neumann architecture. The original work of John von Neumann (von Neumann, 1956), which later became the foundation for modern computers, however, had very little in common to a sequentially executed algorithmic framework. It instead described calculus, boolean logic, and computation with abstracted "neurons" and networks thereof whose intrinsic nature was rather parallel.

tion 1.3). This definition of visual scene understanding is too restrictive since it excludes any form of (dynamic) behaviour which might be generated by a system as a result of its capability to visually understand its environment or scene. The definition of visual scene understanding can be extended to include also that a system understands its visual environment or scene if it is capable of generating behaviour which is appropriate given the visual input and the system's objective (Mysliwetz & Dickmanns, 1986; Dissanayake *et al.,* 2001; Bojarski *et al.,* 2016, & see Section 1.4).

A roadmap to visual scene understanding: While the neuroscientific community agrees that the underlying mechanisms constituting our understanding of scenes needs to be realised using a variety of neuron and synapse types, and also neuromodulation, the nature of the underlying neural code and learning mechanisms are matter of current research and subject to ongoing debate (e.g. Brunel & Wang, 2001; Dayan & Abbott, 2001; Kullmann *et al.,* 2012; Baudot *et al.,* 2013; Zenke *et al.,* 2015; Goel & Buonomano, 2016).
On the quest to enable neuromorphic sensory-processing systems to understand visual scenes and to study to what extent information is present in the precise timing of spikes we need to define a framework of neural algorithms which incorporate mechanisms to extract information from the timing of spikes but also preserve temporal information in their outputs. This will not only provide us with novel algorithms compatible with neuromorphic computing systems, but more importantly, this framework would let us explore the computational advantages of representing time and its implications for visual information processing.
To define such a neural algorithmic framework for visual scene understanding, a first necessary step is to identify the limitations of state-of-the-art neuromorphic sensory-processing systems. A reasonable approach to identify these limitations is to build a neural system which is defined by the network's adjacency matrix to perform the simplest form of visual scene understanding: Reactive collision avoidance behaviour solely based on visual inputs. Even though this simple behaviour has been demonstrated previously already (Sections 2.1 & 3.1), designing such a system enables us to develop ways to compensate for the identified limitations. In a second step, we can extend the previously designed system to incorporate information about the 3D structure of its environment. To estimate the 3D structure, we can use temporal correlations of adjacent inputs (see Sec-

tion 3.1.1) and thus we need to construct a neural algorithm which is capable of estimating temporal correlations. If this algorithm is capable of preserving the estimated temporal correlation in its output, we can introduce an implicit temporal representation of time in ongoing processing. Once we are able to measure temporal correlations in a neural substrate, we finally can design a system which uses the relative timing of incoming (bottom-up) spikes and self-generated (top-down) spikes to perform learning and temporal predictions.
If we assume that a population of spiking neurons learns spatio-temporal patterns from time-continuous inputs, we could use the inherent representation of time in the input to incorporate an implicit representation time in the ongoing processing and time could serve as the necessary "teacher" signal for learning (Mauk & Buonomano, 2004; Karmarkar & Buonomano, 2007; Goel & Buonomano, 2016). The ongoing activity could be used to predict future inputs (Rao & Ballard, 1999; Kilner *et al.,* 2007; Keller & Mrsic-Flogel, 2018) and thus would install a spatio-temporal expectation in ongoing activity. Thus we would be able to address to what extent information about the visual scene is present in the temporal domain and what high-level implications temporal information processing has on our understanding of computation (Goel & Buonomano, 2016).
To test these hypotheses mentioned above, we need to formulate a neurally-implemented algorithm that can learn temporal sequences of spatio-temporally correlated features, and that can generate local, temporally-precise predictions of inputs or internal activity. The temporal correlations of the predicted and incoming activity could thus serve as criteria to gate information transmission and learning. In other words, the gating criterion can be understood as to which extent populations of neurons are in agreement with one another in their spatio-temporal activity. Hence, the definition of appropriate behaviour can be extended: To generate temporally precise predictions of neural activity. If these predictions are temporally precise, i.e. result in a temporal correlation close to 1, the system would have an intrinsic understanding of its input.
This would enable us to study not only predictive scene understanding in SNNs, but also would provide us with a starting point to develop new tools and even better suited neural algorithms to study the benefits and drawbacks of precise temporal information in neural computation.

The structure of this thesis: This thesis aims to enable neuromorphic sensory-processing to understand their visual environment by explicitly extracting information from the temporal domain of their inputs and incorporating an implicit

temporal representation time in their ongoing processing. In Chapter 1 we first discuss two possibilities of how information can be encoded in biological and artificial neural networks, before we provide in Sections 1.1 & 1.2 an overview of available asynchronous sensory-processing systems which can support both information encoding strategies and which are suited to generate real-time behaviour dynamically adapted to the environment. We then provide an overview of relevant algorithms for visual scene understanding operating on frame-based (see Section 1.3) and event-based inputs (see Section 1.4).

We present previous attempts to enable neuromorphic sensory-processing perform visual scene understanding by generating appropriate behaviour in Sections 2.1 and 3.1 and put them into perspective of the prosed solutions here in Sections 2.4 and 3.5, We then present an alternative neuromorphic sensory-processing system to understand its visual environment by learning and predicting precisely timed spatio-temporal patterns. Previous attempts to learning these types of patterns are summarised in Section 1.4 and the proposed solution is put into perspective in Section 4.4. We will provide the reader with a pre- and postamble for Chapters 2, 3 and 4 to put each chapter into perspective of the aim of this thesis and highlight open challenges.

In Chapter 2 we present a hardware friendly reactive collision avoidance control neural algorithm for a simple collision avoidance behaviour. This approach is agnostic to any kind of high-level feature extracted from its sensory input, and it demonstrates how noisy and inhomogeneous processing elements in combination with limited synaptic precision can be compensated using neuronal populations. To equip neuromorphic processing systems with a feature extraction mechanism relevant for collision avoidance, we present in Chapter 3 a novel approach to extract relative motion cues from neuromorphic vision sensors, while the agent is moving through its environment. The proposed spiking Elementary Motion Detector extracts relative motion cues by estimating temporal correlations of adjacent inputs and preserves this temporal correlation information in its output. Finally we will present a SNN which is able to learn spatio-temporal features from event-based sensors (see Chapter 4 & Section 6.2.1). This recurrent SNN is equipped with completely unsupervised learning mechanisms. It has the ability to extract spatio-temporally correlated input events, and exhibits emergent network dynamics which are used to perform spatio-temporal predictions.

We use software simulation to construct neural algorithms which we test in event-based simulations on CPUs. These neural algorithms are characterised by

distributed processing elements which can operate in parallel and operate on asynchronously sampled data. We validate the developed neural algorithms both by emulating them on state of the art mixed-signal sub-threshold neuromorphic processors and by mapping them onto a custom, mixed-signal, sub-threshold neuromorphic test chip.

We discuss each chapter with respect to the introduction of this thesis (see Chapter 5) and conclude this thesis with an overview of the scientific contributions made (see Chapter 6).

1

Time-based computation

THE mammalian nervous system, especially at early processing stages, needs to be subject to internally and externally induced changes to enable perception of its environment (Stevens *et al.,* 1976). In this chapter, we will discuss the benefits and limitations how these changes can be represented by two different modes of operation of neural networks and review briefly event-based sensory and processing systems which can support the identified modes of operation.

A change can be encoded as an event at a precise moment in time. Such events can be integrated by a spiking neuron and generated in the form of an action potential to communicate to other spiking neurons. Spiking neurons can convey information in the form of an analogue value x using discrete events by sampling events from a Poisson process (Troy & Robson, 1992; Softky & Koch, 1993; Dayan & Abbott, 2001), in which the probability of generating an event is proportional to the desired analogue value x. Thus, the instantaneous firing rate $r(t)$ of a neuron is equal to the expected value x. The sampling process, however, has the drawback that the precise time within the fixed time window when an event is generated does not carry information according to the definition of the sampling process, since only on average are x events sampled within a fixed temporal window, and thus information is encoded in a *rate-code*. It is indeed possible to increase the instantaneous firing rate of a neuron within a brief temporal window by increasing its firing probability arbitrarily close to one, thus generating precisely timed events, but with the sampling approach the information conveyed between neurons remains rate-coded.

On the one hand, such a rate-coded representation of information can be an advantage. From a theoretical point of view, one can use probability density functions to draw independent spike times given the instantaneous firing rate,

which carries information for example about the orientation of a stimulus in a continuous manner (Dean, 1981; Dayan & Abbott, 2001). From an experimental point of view, the random process of spike generation matches observations of cortical neurons which seem to fire rather irregularly during stimulus presentation (Softky & Koch, 1993). The response profile of single neurons located in the early visual areas is, since the seminal work of Hubel & Wiesel (1962), typically characterised using relatively simple drifting grating stimuli. Neurons show a trial-to-trial variability in their spiking activity to these type of stimuli which becomes manifested in Poisson spike count (Dean, 1981). These variable responses are commonly interpreted as an independent source of noise inherently present in ongoing processing which is linearly added to the deterministic sensory signal (Baudot *et al.,* 2013), and this linear separation of signal and noise was used to justify rate- and/or population coding by averaging across time and/or neuronal populations (Shadlen & Newsome, 1998; Dayan & Abbott, 2001).

On the other hand, the time when an event is elicited might carry arbitrarily rich information since an underlying algorithm could use the temporal-correlation of events (Dayan & Abbott, 2001; Thorpe, Delorme, *et al.,* 2001), and thus this mode of operation is named *temporal-code*. In the last three decades, more evidence has been provided that spike timing can be useful from theoretical investigations for fast visual processing (Thorpe, Delorme, *et al.,* 2001), classification of time-varying signals and balance (Gütig & Sompolinsky, 2006; Deneve & Machens, 2016), temporal interval discrimination (Buonomano & Merzenich, 1995), state-dependent computation (Buonomano & Maass, 2009) and even motor control (Laje & Buonomano, 2013). Also, experimental findings suggest that the timing of spikes carries information, which can be used by the brain to discriminate textures (Hipp *et al.,* 2006; Wolfe *et al.,* 2008), classify temporal patterns (Mainen & Sejnowski, 1995; Wehr & Zador, 2003; Baudot *et al.,* 2013; Goel & Buonomano, 2016) and even localise oneself in an environment (O'Keefe & Recce, 1993). This evidence highlights that neural networks, whether biological or artificial, can use the temporal information present in ongoing spiking activity to extract relevant cues for behaviour, as well as generate precisely timed events with sub-millisecond variation to time-varying input patterns (Mainen & Sejnowski, 1995). Using intracellular recordings of neurons located in the primary visual cortex of cats Baudot *et al.* (2013) demonstrated that neurons maintained a high temporal precision during sparse activity only in the context of natural images which were animated by simulated eye movements, but not

when more traditional stimuli such as drifting gratings were presented.
It is fair to say that both encoding paradigms, i.e. rate- and temporal-code, convey information in neural networks and actually occur in parallel. However, our current understanding of artificial neural networks is, for the most part, a rate-mode understanding. This is partially due to the structure of the presented inputs, e.g. stationary frames, and partially due to the constraints of the procedure used to train such networks, e.g. the need for a differentiable transfer function and the explicit input-output mapping (e.g. LeCun *et al.,* 2015; Goodfellow *et al.,* 2016, & see Section 1.3 for more details). To expand our understanding of the computational benefits of temporal-coding for the representation of information, we propose the use of neural networks which, besides rate-coding, also convey information in the temporal dimension of their processing. By doing so, we aim to discover novel ways of how information can also be encoded in the relative timing of spikes. Hence, information could potentially be conveyed faster, but more interestingly it could open novel perspectives on spatio-temporal feature and sequence learning, dynamic routing of information based on relative timing and even memory formation, as the system would have an implicit temporal representation of time in ongoing processing. For this purpose an ideal sensory system should be able to sense asynchronously, as well as an ideal downstream processing architecture, should in principle be able to process information asynchronously and in parallel. In the following two sections we will lay out the foundations for asynchronous and fully-parallel sensory systems and alternative computing architectures which can potentially exploit the temporal information introduced by these sensory systems.

1.1 Fundamentals of event-based sensing

Event-based vision sensors are fundamentally different from traditional cameras (see Chapter 3 of Liu, Delbruck, *et al.,* 2014). Contrary to the frame-based approach, event-based sensors typically produce data only when there are changes in the visual scene, which makes them inherently efficient (Lichtsteiner, Posch, *et al.,* 2008; Posch, Matolin & Wohlgenannt, 2010; Brandli *et al.,* 2014; Posch, Serrano-Gotarredona, *et al.,* 2014; Son *et al.,* 2017). The event-driven nature of these sensors reduces redundancy in the data they produce and thus enables higher temporal resolution ($>$ 1 million events per second) (Lichtsteiner, Posch, *et al.,* 2008) compared to standard cameras (Liu, Delbruck, *et al.,* 2014). The Dynamic Vision Sensor (DVS) has an intra-scene dynamic range of 2 lux to 100

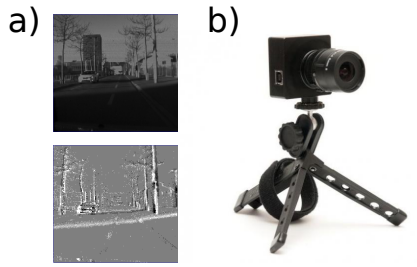


Figure 1.1: The Dynamic Vision Sensor (DVS). a) Comparison of the same scene as seen by a conventional frame-based camera (top) and an event-based DVS camera (bottom). Note that the DVS only sees objects at locations where the temporal contrast changes, e.g. edges. White and black pixels represent "on" or "off" events respectively. To generate the DVS picture, events within a 35 *ms* time-window were accumulated. b) The DVS [Taken from (*iniLabs Ltd* n.d.)].

klux and a power consumption of 23 *m*W (Lichtsteiner, Posch, *et al.,* 2008). These changes can either be passively observed when objects move or actively generated when the sensor is moved through its environment. Another key difference between event-based vision sensors and frame-based ones is that each pixel operates asynchronously and independently of other pixels. In other words, each pixel emits, upon the occurrence of a luminance change, its address in the form of an Address Event Representation (AER) event (Mahowald, 1992; Mahowald, 1994, & see Equation 1.1), if its sensed luminance changed relatively as low as 10 % depending on the sensor settings. The AER protocol is an asynchronous handshaking protocol, where each event is written onto a common transmission bus, which is shared by all pixels on the chip (see Chapter 2 in Liu, Delbruck, *et al.,* 2014). If we consider an event generated at time t with a polarity pol and a spatial location x, y we can write:

$$e = (x, y, t, pol), \tag{1.1}$$

where x, y indicates the spatial coordinate in the pixel array, pol the sign of the change (positive (on) or negative (off) relative change). In these sensors time represents itself. The time-stamp t is only attached by a Field Programmable Gate Array (FPGA) after the luminance change was detected to allow conventional processing systems to visualise and operate on this type of data. It is worth stressing that these events are characterised primarily by the time of production and only secondarily by their inherently encoded payload. As depicted in Equation 1.1 the payload of such events is the pixel's location and the sign

of the detected temporal contrast change. However, an event could also carry additional payload information, such as the orientation of a detected edge, relative motion or in fact the presence of any spatio-temporal pattern (e.g. *The jAER Open Source Project* 2007). A key feature of event-based sensors is that events occur at precise times, meaning that if we want to design an event-based algorithm, we need to consider the temporal information during the execution of the algorithm. This requires not only that the algorithm should extract meaning from the temporal information of incoming events, but also opens novel ways to express error signals in the temporal dimension, as we will show and discuss later (see Chapter 4 & Section 6.2.1). As the nature of information in a stream of events is partially encoded in the temporal domain, an algorithm operating on this asynchronous stream should be capable of incorporating and conveying this temporal information. Examples of event-based algorithms, e.g. for edge detection or estimation of optic flow, could be executed on conventional processing architectures by defining appropriate interrupts to reflect the asynchronous nature of event-based data. If, on the other hand, we design these algorithms for dedicated asynchronous parallel processing hardware architectures, we do not need to define interrupts to trigger certain computations, but we are still able to extract this information. More interestingly, we might observe emerging computational properties, such as the preservation of temporal information in the output of the implemented algorithm (see Chapter 3). The asynchronous and parallel sensing paradigm not only strictly constrains the way an algorithm has to be designed (see Section 1.4), but it also dictates a processing paradigm which uses an implicit temporal representation of time in ongoing processing.

1.2 Fundamentals of event-based computation

To perform computation on event-based inputs, we can either use conventional processing architectures and optimise the execution routines to fit the unconventional data type, or we use unconventional, neuromorphic processing architectures which inherently feature event-driven, asynchronous communication and parallel computation. Conventional processing architectures, like the classical von Neumann CPU architectures, are characterised by the separation of memory and computation (Backus, 1978). Before an algorithm can be executed the data, e.g. an image and algorithmic parameters, needs to be loaded from peripheral memory into main memory and finally into the cache memory of the CPU (see Figure 1.2). During the transfer of the data, the processor is idle while data

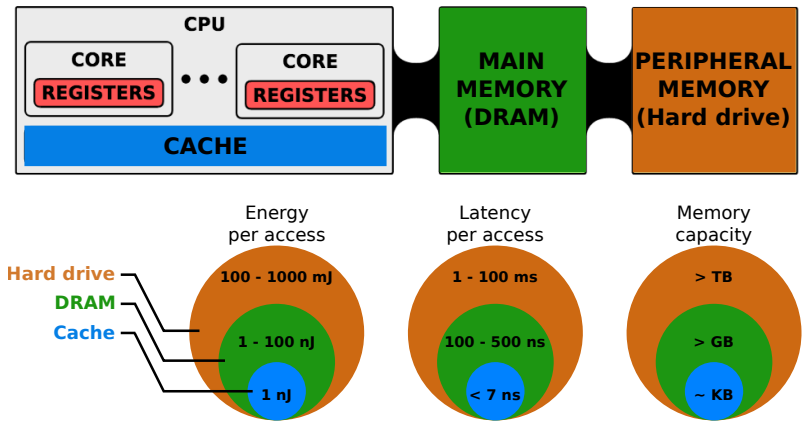


Figure 1.2: The von Neumann bottleneck. Memory arrangement and different layers of memory in classical von Neumann architectures. Comparison is done in terms of energy consumption, latency and memory capacity. Figure taken from (Milde, Neil, *et al.,* 2017).

is transferred between different levels of memory and into the CPU. After the image is loaded all defined computations are executed.

Assume that the algorithm should detect oriented edges and assume further that the image at hand only consists of vertically oriented bars. A conventional algorithm would convolve the image with all possible oriented edge detectors at all possible positions, only to report that most of its convolutions are in fact close to zero. Furthermore, if a second image, e.g. from a video stream, is to be analysed and the image does not change from one frame to the next, a conventional algorithm executed on a conventional processing architecture would waste the same computational resources, only to report the same result for each frame[1].

Although tremendous efforts have been made to design special purpose hardware to reduce time the processor remains idle by optimising memory transfer, incorporating efficient compression schemes (Conti *et al.,* 2015; Cavigelli & Benini, 2017; Aimar *et al.,* 2018) and skipping computations if the input or the neuron's activity is close to zero (Aimar *et al.,* 2018), the underlying clock-based logic of conventional processors and algorithms result in non-optimal processing in the context of time-continuous event-based data, since each computation is

[1] Graphics Processing Units (GPUs) can certainly parallelise computation (Nvidia, 2007), however, in a hierarchical processing scheme such as multilayer ANNs all defined computations need to be finished in a given layer (depending on the available Synchronous Dynamic Random Access Memory (SDRAM)), before the next layer can be computed.

executed at a pre-defined clock frequency. There exist solutions to construct multi-clock systems which depending on the data density run at a higher clock frequency with the drawback of a higher energy consumption (Liu, Bellec, *et al.,* 2018).

Some non-von Neumann architectures, e.g. distributed processing architectures with co-localised memory and computation, offer a promising alternative for event-based processing. These types of processors can be summarised by the term neuromorphic processors. Proposed neuromorphic processors range from synchronous, multi-chip systems (Furber, Lester, *et al.,* 2012; Furber, Galluppi, *et al.,* 2014), fully digital, synchronous FPGA-based systems (Wang & van Schaik, 2018), digital, asynchronous time-multiplexed systems (Akopyan *et al.,* 2015; Davies *et al.,* 2018), to mixed-signal analogue/digital sub-threshold, fully asynchronous space-multiplexed systems (Benjamin *et al.,* 2014; Qiao *et al.,* 2015; Moradi *et al.,* 2018; Neckar *et al.,* 2019). For a detailed review on existing solutions, please refer to Thakur *et al.* (2018).

The mixed-signal analogue/digital sub-threshold, fully asynchronous space-multiplexed processing scheme represents an exciting processing alternative regarding event-based inputs. These systems are characterised by physically repeated, complementary metal-oxide-semiconductor (CMOS) circuits that emulate certain characteristics and biophysical properties of biological neural networks (Mead, 1989; Liu, Kramer, *et al.,* 2002; Chicca *et al.,* 2014). This physical repetition of computing elements, e.g. neurons and synapses, allows these processors to operate in a fully parallel, asynchronous manner and in real-time (Liu, Delbruck, *et al.,* 2014). Although this seems like the holy grail of processing event-based data, sub-threshold neuromorphic processors have severe constraints. Firstly, even though each computing element is fabricated according to the same schematics and repeated on the silicon wafer, variations in the concentrations of dopants across the wafer and imperfections in the manufacturing process result in device mismatch which leads to variable output functions given the same input (Pelgrom *et al.,* 1989). The circuit to circuit variability is usually around 20%. Secondly, due to limitations on the physical size of a system implemented on a silicon wafer, as area is quite expensive and because silicon synapses are comparatively large on area when compared to the size of the remainder of a silicon neuron, the total number of synaptic inputs to any given neuron is limited in particular systems (Moradi *et al.,* 2018). While there exist sub-threshold neuromorphic processor solutions which feature an all-to-all connectivity scheme (Qiao *et al.,* 2015), this is only feasible for small scale

networks of up to 256 neurons[2], but not on a larger scale network of neurons consisting of thousands or millions of neurons. Thirdly, when one connects two neurons with a synapse, the associated synaptic weight can in some particular systems, such as the Reconfigurable On-Line Learning System (ROLLS) and the Dynamic Neuromorphic Asynchronuous Processor (DYNAP), be set to take a value with 12-bit resolution (Qiao *et al.,* 2015; Moradi *et al.,* 2018). This does not sound like a hard constraint, however, the synaptic weight which is set by a bias voltage in the aforementioned neuromorphic processors is shared across 256 neurons and can only take on one out of four absolute levels for the subset of neurons (two absolute levels for excitatory and two absolute levels for inhibitory weights). That has the consequence that the effective weight precision is two bits.
We will demonstrate how device mismatch can be compensated by population averaging and how the limited synaptic weight resolution can be compensated by using multiple synaptic connections between two neuronal populations (see Chapter 2). We will further demonstrate how synaptic efficacy can be adaptively adjusted to extract and convey temporal information in ongoing processing (see Chapter 3) and ultimately how mismatch can actually be used for stable spatio-temporal feature learning and spatio-temporal prediction of activity (see Chapter 4).

We discussed what the fundamental characteristics of event-based sensory-processing systems and to what extent these systems are different from their clock-based counterparts. What we have not addressed yet is in which contexts, i.e. tasks, the event-driven processing paradigm could potentially be a promising alternative. Before we can go into the detail of visual scene understanding in neuromorphic sensory-processing systems, we need first to understand what the limitations and drawbacks of conventional approaches to visual scene understanding are.

[2]Crossbar array architectures (Kim *et al.,* 2012; Truong *et al.,* 2014; Mostafa, Mayr, *et al.,* 2016; Zidan *et al.,* 2016) which replace synaptic circuits with memristive devices (Chua, 1971; Chua & Kang, 1976) represent a promising, scalable alternative which could potentially feature all-to-all connectivity of arbitrary size. These systems and their applications are subject to ongoing research (Truong *et al.,* 2014; Payvand, Muller, *et al.,* 2018; Payvand, Nair, *et al.,* 2018).

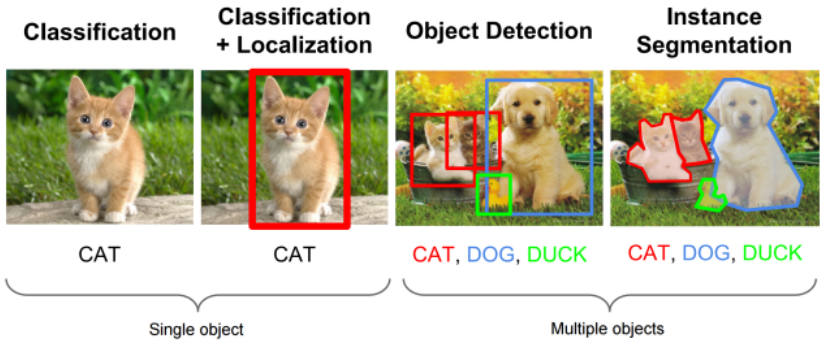


Figure 1.3: Visual scene understanding in Artificial Neural Networks. Figure taken from (dos Santos, 2018).

1.3 Visual scene understanding in Artificial Neural Networks

Visual scene understanding is commonly defined as identifying objects of interest in an image (Krizhevsky *et al.,* 2012; He *et al.,* 2016), indicating their location (Girshick, 2015; Ren *et al.,* 2015) and segmenting these images on a per pixel basis to label each pixel as belonging to a certain category, such as "dog", "street" or "side-walk" (Girshick *et al.,* 2014; Long *et al.,* 2015). See Figure 1.3 for examples. To train ANNs to classify visual inputs techniques, such as error-backpropagation (Rumelhart *et al.,* 1986) is used in combination with an optimisation procedure (Cauchy, 1847; Robbins & Monro, 1951; Bryson & Denham, 1962). To this aim ANNs are presented with a tremendous amount of hand-annotated training-data, consisting of input images and desired output-vectors. This trainings procedure is referred to as *supervised learning*. The input-output dimensionality is fixed and only in limited situations and particular architectures expandable (Shmelkov *et al.,* 2017). As each input data sample is labelled, the objective is to minimise a pre-defined loss function by optimising the network's parameters (Robbins & Monro, 1951). During the optimisation, the global error produced by the network is differentiated with respect to the neuron's weights (Robbins & Monro, 1951; LeCun *et al.,* 2015). This can be understood as a credit assignment for each neuron which took part in generating the error (LeCun *et al.,* 2015). The resulting update of each parameter, i.e. synaptic weight, is proportional to how much it contributed to the error. As the network's parameters are optimised by minimising the task-specific loss function, each objective, e.g. localising an object's position or object's identity, needs its own labelled data set. This means that a network whose objective is to identify the position of an object in an

image needs a network which identifies the object (Ren *et al.,* 2015). This also implies that the network whose task is to identify objects in an image does not represent the object's position (Krizhevsky *et al.,* 2012). It seems appealing to assign task-specific functions to each sub-network explicitly, but the explicit loss function formulation prevents the system from re-using computational resources which are, ultimately, limited (for more information on transfer learning, please refer to Thrun & Pratt, 2012). Thus conventional approaches to visual scene understanding using ANNs are in their essence input-output function approximations which are stacked together and trained, in some cases independently of one another (Girshick, 2015; Ren *et al.,* 2015). The generalisation of these networks, which is defined as the percentage of correctly classified inputs sampled from unseen test data, is task-specific (Zhang *et al.,* 2016). Meaning that a network which can determine the position and identity of a pedestrian, can not do the same identification process for an animal if it was not explicitly part of the training set and thus not incorporated in its parameter optimisation.

While learning to map pre-defined input-output functions, in fact, meets the definition of scene understanding mentioned above, it is also inherently limited to this definition. This is because the network is trained based on the network's global, task-specific error and any parameter update is non-local, thus affecting the entire network performance. The local optimum found by gradient descent is a local optimum given the amount of data, the loss function, and the task (Gori & Tesi, 1992). As soon as we extend the definition of scene understanding to include the depth structure of the environment, a new loss function needs to be defined and new local minimum needs to be found, even though the information already available about the relative position of objects and their identity might be beneficial for any of the tasks (Pratt & Jennings, 1996; Thrun & Pratt, 2012). Despite recent advances in making the training process more biologically plausible, e.g. using feedback alignment (Lillicrap *et al.,* 2016; Neftci, Augustine, *et al.,* 2017), spikeprop (Bohte *et al.,* 2002), targetprop (Bengio, Lee, *et al.,* 2015) or layer-wise, greedy training (Bengio, Lamblin, *et al.,* 2007), the network's single objective is to generate an output in a pure feedforward way and the implemented feedback is to only propagate a global error for the single objective of interest, e.g. object classification. The recent advances in terms of classification accuracy and task complexity using ANNs are rather due to more computing resources (Nvidia, 2016), better programming interfaces with optimised vector-matrix multiplications (Jia *et al.,* 2014; Martín Abadi *et al.,* 2016) and of course more labelled data, than to a fundamental shift in the way these

networks are trained (Jordan & Mitchell, 2015).
Autoencoders represent a promising alternative in the case of *unsupervised learning* and input-driven visual scene understanding in ANNs (Hinton & Zemel, 1994). The idea of autoencoders is rather simple: Given a certain set of inputs find a set weights which project the input space into an embedding space and use the activity therein to reconstruct the input. This objective is fulfilled when

$$\mathrm{W}_{encoding} = \mathrm{W}^{\mathrm{T}}_{decoding}, \tag{1.2}$$

where $\mathrm{W}_{encoding}$ is the weight matrix projecting the input space into the embedding space and $\mathrm{W}_{decoding}$ is the weight matrix projecting the embedding space back to the input space, where $^{\mathrm{T}}$ denotes the transpose operator. The embedding or feature space can either be a lower dimensional space, i.e. compressing autoencoders, or a higher dimensional space, i.e. expanding autoencoders, compared to the input dimensionality (Vincent, Larochelle, Bengio, *et al.,* 2008). The input is perfectly reconstructed if the encoding weight matrix is the transpose of the decoding weight matrix (see Equation 1.2). Autoencoders were shown to be capable to reconstruct a given input even in the presence of noisy inputs for handwritten digit recognition (Vincent, Larochelle, Bengio, *et al.,* 2008; Vincent, Larochelle, Lajoie, *et al.,* 2010), object recognition (Masci *et al.,* 2011) and auditory classification (Vincent, Larochelle, Lajoie, *et al.,* 2010).
If we stop distinguishing the network's activity between training and testing, thus building "online" networks which continuously process presented inputs and generated outputs, we could try to define a local error, similar to stacked autoencoders (Vincent, Larochelle, Lajoie, *et al.,* 2010; Shin *et al.,* 2013). In the context of a SNN this error could use the timing of incoming inputs and generated output as some sort of "teacher" signal and thus have locally available error information. This would loosen the boundaries of fixed input-output pairs in training and might provide us with novel insights on learning in general.

1.4 Visual scene understanding in Spiking Neural Networks

In this line of thought we can also state that an agent, i.e. a robot, or a network understands its visual input if it is capable of generating appropriate behaviour given its visual input and the agent's objective, e.g. avoiding collisions with objects while navigating (Mysliwetz & Dickmanns, 1986; Dissanayake *et al.,* 2001; Bojarski *et al.,* 2016), able to make correct (temporal) predictions about the per-

ceived visual inputs (Hinton & Zemel, 1994; Vincent, Larochelle, Bengio, *et al.,* 2008; Vincent, Larochelle, Lajoie, *et al.,* 2010; Lotter *et al.,* 2016) or capable of predicting ongoing neural activity. The examples mentioned above use either single frames or successive frames of a video as their visual input. In both cases, the provided input data, i.e. frames, is sampled synchronously (see Section 1.1) If we want to present asynchronously sampled data to a conventional ANN we either need to transform the asynchronous stream of events into a synchronous, frame-like input (Lungu *et al.,* 2017; Gallego *et al.,* 2018; Yousefzadeh *et al.,* 2018, & see an event-frame example in Figure 1.1), or rely on an event friendly ANN architecture like phased-Long Short Term Memory (LSTM) (Neil *et al.,* 2016).
Alternatively, we can present the asynchronously sampled data to a SNN. SNNs are not better suited than ANNs to understand visual scenes, however, in the context of event-based data, SNNs have in their essence one fundamental advantage compared to ANNs: SNNs can, in principle, incorporate continuous time naturally in their ongoing processing. The reason that SNNs are not the primary choice when it comes to event-based visual processing is twofold: (i) they are difficult to train due to missing unified theoretical framework and software tools and (ii) SNNs do not outperform ANNs in terms of classification accuracy[3]. The former is mostly due to the fact that existing methods to train ANNs cannot be directly mapped to SNNs as each spike represents a discontinuity in the transfer function which thus cannot be differentiated. Additionally, the computational advantage of transmitting information using spikes is diminished, because to reach a similar performance, compared to standard ANNs, the network would need to integrate spikes for a longer time (Rueckauer, Lungu, *et al.,* 2017). The latter is because we have not yet thought of a task in which frame-based sensing and processing is actually disadvantageous from a computational point of view.

If we want to use SNNs in the context of event-based visual scene understanding in spite of these drawbacks we could (i) create a new labelled data set using event-based sensors and train a conventional ANN and convert the trained ANN into an SNN (Lungu *et al.,* 2017; Rueckauer, Lungu, *et al.,* 2017; Stromatias, Soto, *et al.,* 2017; Yousefzadeh *et al.,* 2018; Bichler, Briand, *et al.,* 2019), or (ii) train an SNN directly with feedback alignment (Neftci, Augustine, *et al.,* 2017) or using dual accumulator neuron models (Thiele *et al.,* 2018). In both

[3]If we would expand the figure of merit to include power consumption per operation, latency and computing time SNNs by far outperform ANNs.

cases mentioned above, the need for labels is inevitable. In the first case, the situation is even worse, because the input is a pixel-averaged event-rate which is sampled as an instantaneous rate from a Poisson distribution, and so precise temporal information is lost. We could (iii) design a task-specific, hard-coded network architecture to operate on a stream of events in order to navigate in unknown environments (Indiveri & Verschure, 1997; Maris & Mahowald, 1997; Etienne-Cummings *et al.,* 1999, and see Chapter 2 & 3). Or we could (iv) learn spatio-temporal templates which best match incoming spatio-temporal patterns (Bichler, Querlioz, *et al.,* 2012; Lagorce, Orchard, *et al.,* 2017; Afshar *et al.,* 2019, and see Chapter 4).

The interesting aspect about learning spatio-temporal templates is that these "filters" operate in both the spatial (x, y) and in the temporal dimension. The idea is rather simple: for any incoming event, its neighbouring events in a local spatial neighbourhood are selected (Lagorce, Orchard, *et al.,* 2017). Each event in this selected neighbourhood is convolved with an exponentially decaying kernel with a constant temporal decay. The event in the centre of the spatial neighbourhood is used as a reference point in time for the exponentially decayed kernels. The resulting values of the decayed kernels at the time of the event in the centre represent a so-called time surface (Lagorce, Orchard, *et al.,* 2017, & see Figure 2 therein). A time surface can be understood as to filter each event using a synapse which follows a first-order differential equation (see Equation 3.3 as an example). From a pre-defined, randomly initialised set of feature detectors, the one feature detector which shows the highest similarity, or smallest distance, to the current time surface is selected. The cosine distance between the feature detector's weight matrix and the time surface is calculated and used to update the weights (Lagorce, Orchard, *et al.,* 2017). This procedure is repeated for every incoming event until either all events are processed (Lagorce, Orchard, *et al.,* 2017) or the weight values of the feature detectors do not change any more (Afshar *et al.,* 2019). A rather biologically implausible constraint of these approaches is that only one feature detector can update its weights at any given time (Lagorce, Orchard, *et al.,* 2017; Afshar *et al.,* 2019). In other words, this is like an instantaneous inhibition of other feature detectors and can be understood as a competitive mechanism. To guarantee that all feature detectors learn equally Afshar *et al.* (2019) introduced an adaptive thresholding mechanism, which has the effect that once a feature detector is updated, all non-updated detectors are more likely to be updated for the next time surface. Both of these approaches can be understood as template matching, in which information is

processed in a feedforward way and "lateral" interactions only serve a competitive purpose. Furthermore, neither Lagorce, Orchard, *et al.* (2017) nor Afshar *et al.* (2019) provide a neuronal implementation of their algorithm, making it difficult to judge whether these algorithms could be emulated on a neuromorphic processor. An interesting solution to this was presented by Bichler, Querlioz, *et al.* (2012), where the authors propose a neural architecture which learns based on the well-known local Spike-Time Dependent Plasticity (STDP) learning rule to track the spatio-temporal features and position of a moving ball. The SNN was simulated in software and performed all computations off-line. The implemented STDP protocol has an infinitely long depression window which and the lateral connections between the learned feature detectors are also only used to introduce competition among the feature detectors (Bichler, Querlioz, *et al.,* 2012).

To leverage the full potential of SNNs, we need to study the effect of lateral and top-down feedback connections, which not only introduce competition but also convey useful information to learn spatio-temporal features, such as sequences which represent temporal expectations in ongoing processing.

1.5 Conclusion

Event-based sensors and processors which can emulate SNNs have an additional computationally meaningful dimension: time (see Section 1.1 & 1.2). This extra dimension opens new perspectives on how computation could be achieved and how information can be transmitted in a neural network. We briefly presented theoretical and experimental evidence that *rate-codes* and *temporal-codes* can coexist in biological and artificial spiking neural networks (see beginning of Chapter 1). Furthermore, by discussing the training process of ANNs, we highlighted that conventional sensory and processing systems cannot capture these temporal aspects of computation and information transmission as these networks are based on *rate-codes* (see Section 1.3). While there exist event-based solutions which do respect the temporal neighbourhood of an event, these solutions are either not suited for neuromorphic processors or only use lateral feedback for inhibition (Bichler, Querlioz, *et al.,* 2012; Lagorce, Orchard, *et al.,* 2017; Thiele *et al.,* 2018; Afshar *et al.,* 2019). In order to enable SNNs to understand visual scenes we first will build a system which is capable of performing simple but appropriate behaviour in the context of autonomous navigation (see Section 2). Then we will extend our system to extract information about the relative 3D

structure of its environment by computing temporal correlations to estimate relative motion cues to scale its avoidance behaviour (see Section 3).

PART II

Towards scene understanding

2

Reactive control

Preamble

THE following work not only represents a proof of concept for real-time behaviour generated with mixed-signal neuromorphic technology, but also addresses three fundamental limitations we, as a community, are facing when working with neuromorphic systems: (i) device mismatch, (ii) limited weight resolution and (iii) events generated by ego-motion (for more details see Section 1.2). To address the aforementioned limitations we decided to build a simple neuromorphic Braitenberg vehicle-like system, which avoids in real-time and closed-loop obstacles while navigating in its environment.

Although collision avoidance itself has been demonstrated in a myriad of contexts, i.e. indoors (Denk *et al.,* 2013), outdoors (Milde, Bertrand, Benosman, *et al.,* 2015; Hwu, Isbell, *et al.,* 2017; Hwu, Krichmar, *et al.,* 2017), driving (Milde, Bertrand, Benosman, *et al.,* 2015) or flying (Zingg *et al.,* 2010; Fu *et al.,* 2014), and with a variety of different sensory-processing systems, i.e. CPU, GPU, μcontroller (Conradt, Galluppi, *et al.,* 2015; Luber *et al.,* 2015), light-sensors (Braitenberg, 1986), CMOS cameras (Hwu, Isbell, *et al.,* 2017; Hwu, Krichmar, *et al.,* 2017) or event-based vision sensors (Denk *et al.,* 2013; Milde, Bertrand, Benosman, *et al.,* 2015), solutions proposed so far are often characterised by a mix of neuromorphic sensing elements and conventional clock-based processing or vice-versa. Fully asynchronous solutions using event-based vision sensors and neuromorphic processors were shown to work with fully custom designed processing circuitry, which combined sensing and processing on the same chip (Kramer *et al.,* 1995; Indiveri & Verschure, 1997; Kramer *et al.,* 1997; Maris & Mahowald, 1997). The combination of sensory detection and information

processing circuits not only limits the applicability, as any new sensory system would need new processing circuitry, but it also hinders the reproducibility of the designed algorithm. But more importantly, if we would separate sensing and processing and would design the neuronal algorithm independently of the exact details of the sensing and processing structures, we would be able to more easily apply the same algorithm to different variants of event-based sensors and processors. The algorithm in principles should to able to operate on asynchronous events, whether they are emitted from a DVS (Lichtsteiner, Posch, *et al.,* 2008) or an Asynchronous Time-based Image Sensor (ATIS) (Posch, Matolin & Wohlgenannt, 2010), and needs to be defined by an adjacency matrix of interconnected neuronal populations, whether these populations are emulated on a DYNAP (Moradi *et al.,* 2018) or simulated on a SpiNNaker board (Furber, Galluppi, *et al.,* 2014).

The following chapter was published as Milde, M. B., Blum, H., Dietmüller, A., Sumislawska, D., Conradt, J., Indiveri, G. & Sandamirskaya, Y. Obstacle Avoidance and Target Acquisition for Robot Navigation Using a Mixed Signal Analog/Digital Neuromorphic Processing System. *Frontiers in Neurorobotics* **11,** 28. ISSN: 1662-5218 (2017) and partially adapted for this thesis.

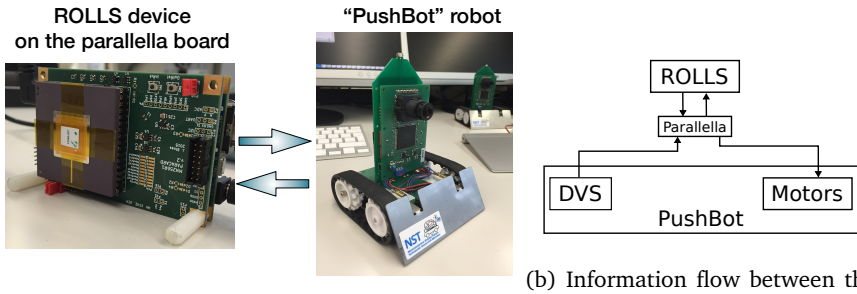
2.1 Introduction

Collision avoidance is one of the most basic tasks in mobile robotics that ensures safety of the robotic platform, as well as the objects and users around it. Biological neural processing systems, including relatively small ones such as those of insects, are impressive in their ability to avoid obstacles robustly at high speeds in complex dynamical environments. Relatively simple neuronal architectures have already been proposed to implement robust obstacle avoidance, e.g. (Blanchard *et al.,* 2000; Iida, 2001; Rind & Santer, 2004), while probably the most simple conceptual formulation of a neuronal controller for obstacle avoidance is the famous Braitenberg vehicle (Braitenberg, 1986). When such *neuronal* control architectures are implemented on a conventional computer, analog sensor signals are converted and stored in digital variables. A large number of numerical computations are performed then, which are required to model the involved neuronal dynamics in software. Neuromorphic hardware offers a physical computational substrate for directly emulating such neuronal architectures in real time (Indiveri, Chicca, *et al.,* 2009; Furber, Lester, *et al.,* 2012; Benjamin *et al.,* 2014; Chicca *et al.,* 2014), enabling low latency and massively parallel, event-based computation. Neuromorphic electronic circuits can implement dynamics of neurons and synapses using digital (Furber, Lester, *et al.,* 2012) or analog (Benjamin *et al.,* 2014; Qiao *et al.,* 2015) designs and allow for arbitrary connectivity between artificial neurons. The analog implementations of artificial neural networks are particularly promising, due to their potential smaller size and lower power consumption figures than digital systems (for a review see Indiveri, Linares-Barranco, *et al.,* 2011; Hasler & Marr, 2013). But these features come at a price of precision and reliability. Indeed, with analog designs, the device mismatch effects (i.e. variation in properties of artificial neurons across the device) have to be taken into account for the development of robust functional architectures (Neftci, Chicca, *et al.,* 2011). A promising strategy for taking these issues into account is to implement the mechanisms used in *biological neural networks*, which face the same problem of using an unreliable computing substrate that consists of noisy neurons and synapses driven by stochastic biological and diffusion processes. These biological mechanisms include adaptation and learning, but also using *population coding* (Ermentrout, 1998; Pouget *et al.,* 2000; Averbeck *et al.,* 2006) and *recurrent connections* (Wilson & Cowan, 1973; Douglas, Koch, *et al.,* 1995) to stabilise behaviourally relevant decisions and states against neuronal and sensory noise. In this work, we show that by using

the population-coding strategy in a mixed signal analog/digital neuromorphic hardware, it is possible to cope with the variability of its analog circuits and to produce reliably the desired behaviour on a robot.

We present a first proof of concept implementation of such a neuromorphic approach to robot navigation. Specifically, we demonstrate a *reactive vision-based* obstacle avoidance strategy using a neurally-inspired event-based Dynamic Vision Sensor (DVS) (Lichtsteiner, Posch, *et al.,* 2006) and a ROLLS neuromorphic processor (Qiao *et al.,* 2015). The proposed architecture is event-driven and uses the neural populations on the ROLLS device to determine the steering direction and speed of the robot based on the events produced by the DVS. In the development phase, we use a miniature computer Parallella[1] solely to manage the traffic of events (spikes) between the neuromorphic devices, and to store and visualise data from the experiments. The Parallella board can be removed from the behavioural loop in target applications, leading to a purely neuromorphic implementation. In this paper, we demonstrate the robustness and limits of our system in a number of experiments with the small robotic vehicle "Pushbot"[2] in a robotic arena, as well as in an unstructured office environment. Several neuromorphic controllers for robots were developed in the recent years, e.g. a SpiNNacker system (Furber, Lester, *et al.,* 2012) was used to learn sensory-motor associations with robots (Conradt, Galluppi, *et al.,* 2015; Stewart *et al.,* 2016), a neural-array integrated circuit was used to plan routes in a known environment (Koziol *et al.,* 2014), three populations of analog low-power subthresold VLSI integrate-and-fire neurons were employed to control a robotic arm (Perez-Peña, Morgado-Estevez, *et al.,* 2013). Our system goes along similar lines and realises a reactive robot navigation controller that uses a mixed signal analog/digital approach, and exploits the features of the ROLLS neuromorphic processor. In this work we follow a dynamical systems – attractor dynamics – approach to robot navigation (Bicho, Mallet, *et al.,* 2000), which formalises one of the famous Braitenberg vehicles (Braitenberg, 1986). The neuronal architecture in our work is realised using a number of neuronal populations on the neuromorphic device ROLLS. The dynamical properties of neuronal populations and their interconnectivity allow to process a large amount of sensory signals in parallel, detecting the most salient signals and stabilising these detection decisions in order to generate robustly closed-loop behaviour in real-world unstructured and noisy environments (Sandamirskaya, 2014; Indiveri & Liu,

[1] https://www.parallella.org
[2] http://inilabs.com/products/pushbot



(a) The hardware setup used in this work: the neuromorphic processor ROLLS is interfaced to a miniature computer Parallella, which communicates with the Pushbot robot over a dedicated WiFi network.

(b) Information flow between the three main components: ROLLS, Parallella, and the Pushbot, in particular, its sensor DVS and two motors.

Figure 2.1

2015). Here, we demonstrate the feasibility of deployment of a neuromorphic processor for the closed loop reactive control. We found several limitations of the simple Braitenberg-vehicle approach and suggest extensions of the simple architecture that solve these problems, leading to robust obstacle avoidance and target acquisition in our robotic setup.

2.2 Materials and Methods

The experimental setup used in this work consists of the Pushbot robotic vehicle with an embedded Dynamic Vision Sensor (eDVS) camera and the ROLLS neuromorphic processor. A miniature computing board Parallella is used to direct the flow of events between the robot and the ROLLS. Figure 2.1a shows the components of our hardware setup, while Figure 2.1b shows the information flow between different hardware components. The Pushbot communicates with the Parallella board via a wireless interface for receiving motor commands and for sending address-events produced by the DVS. Using a dedicated WiFi network, we achieve communication latency below 10ms, which was enough to demonstrate functionality of our system at speeds, possible with the Pushbot. The ROLLS device is interfaced to the Parallella board using an embedded FPGA, which is used to configure the neural network connectivity on the chip and to direct stimulating events to neurons and synapses in real time. The Parallella board runs a simple program that manages the stream of events between the neuromorphic processor and the robot.

2.2.1 The ROLLS neuromorphic processor

The ROLLS is a mixed signal analog/digital neuromorphic chip (Qiao *et al.,* 2015) that comprises 256 spiking silicon neurons, implemented using analog electronic circuits which can express biologically plausible neural dynamics. The neurons can be configured to be fully connected with three sets of synaptic connections: an array of 256x256 non-plastic ("programmable") synapses, 256 plastic ("learnable") synapses that realise a variant of the Spike-Time Dependent Plasticity (STDP) rule (Mitra, Fusi, *et al.,* 2009), and 4 additional ("virtual") synapses that can be used to receive external inputs. In this work, only the programmable synapses were used for setting up the neuronal control architecture, as no online-learning was employed for the navigation task. Figure 2.2 shows a block diagram of the ROLLS device, in which 256 spiking neurons, implemented using analog electronic circuits Indiveri, Chicca, *et al.,* 2006, are shown as triangles on the right, and 256x256 non-plastic ("programmable") synapses, which can be used to create a neuronal architecture on the ROLLS, as well as 256 "virtual" synapses used to stimulate neurons externally, are shown as white squares. A digital Address Event Representation (AER) circuitry allows to stimulate neurons and synapses on the chip, as well as to read-out spike events off chip; a temperature-compensated digital bias-generator allows to control parameters of analog electronic neurons and synapses, such as the refractory period or membrane time constant. The programmable synapses share a set of biases that determine their weight values, their activation threshold, and time constants. These three parameters determine the synaptic strength and dynamics of the respective connection between two neurons. A structural limitation of the hardware is that each synapse can only assume one of eight possible weight values (four excitatory and four inhibitory values). This means that in a neuronal architecture, several different populations might have to share weights, which limits the complexity of the architecture. ROLLS consumes approx. 4mW of power in typical experiments, run here. The ROLLS parameters (biases) used in this work are listed in the Appendix (Section 7.1.2).

2.2.2 The DVS camera

For the obstacle avoidance scenario, important properties of the DVS are its low data rate, high dynamic range, and small sensitivity to lighting conditions (Lichtsteiner, Delbruck, *et al.,* 2006, & see Section 1.1 for overview on event-based sensing.). The challenges are noise, inherent in the sensor, its inability to

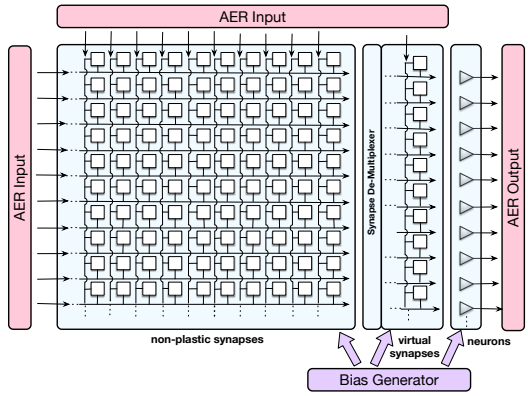


Figure 2.2: The schematic visualisation of neurons (grey triangles), non-plastic and virtual synapses (white squares), as well as input-output interfaces and bias generator of the ROLLS chip. Each neuron on the chip (presynaptic neuron) sends its output spikes to 256 non-plastic synapses, which, if set active, can route these spikes to any of the neurons on the chip (postsynaptic neurons). The connectivity matrix allows for all-to-all connectivity, but also other configurations. AER is a digital Address-Event Representation, used to communicate spikes (it consists of an index of the spike-emitting neuron.)

detect homogeneous surfaces, and relatively small spatial resolution (128x128 pixels), as well as a limited field of view (60°). New versions of DVS are currently available, which would further improve performance of the system. Moreover, more sophisticated object-detection algorithms for DVS are currently being developed (Moeys *et al.,* 2016). The embedded version of the DVS eDVS camera (Müller & Conradt, 2011) used in this work uses an ARM Cortex microcontroller to initialises the DVS, capture events, send them to the wireless network, and to receive and process commands for motor control of the Pushbot.

2.2.3 Neuromorphic Robot

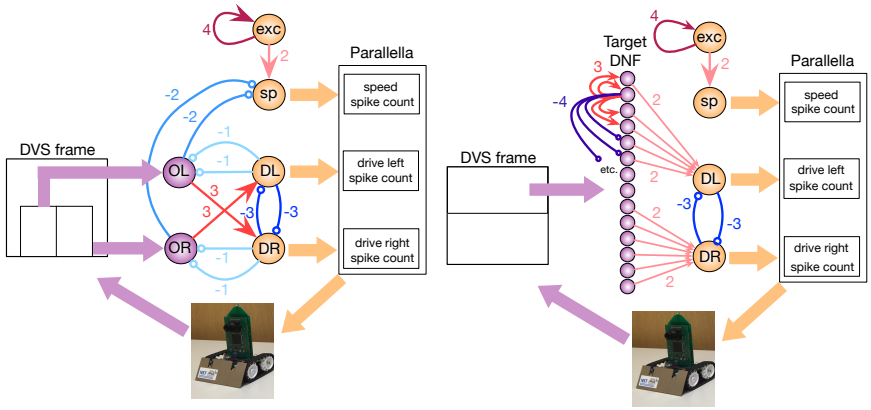
The robot used in this work is the mobile autonomous platform Pushbot, which consists of a $10 \times 10cm$ chassis with two motors driving two independent tracks for propulsion (left and right). The predominant component on the small robot is an eDVS (see Section 1.1), which acquires and provides sensory information and controls actuator output, including the robot's motors, through its embedded microcontroller. The sensor's integrated 9 DOF IMU reports changes of velocity and orientation. The robot actuators include a buzzer, two parallel, horizontal forward laser pointers and an LED on top, which all can show arbitrary activation

patterns. The Pushbot is powered by 4 AA-batteries, which ensure ∼2h operation time. The robot communicates through WLAN at up to 12Mbps, which allows remote reading of sensory data (including events from the eDVS) and setting velocities with a latency $< 10ms$. The Pushbot robot is too small to carry the current experimental hardware setup. In principle, however, it is possible to place the ROLLS chip directly on a robot, removing the WiFi latency.

2.2.4 Spiking Neural Network Architecture

The core of the system presented here is a simple neural network architecture that is realised in the ROLLS device and allows the robot to avoid obstacles and approach a simple target. The "connectionist" scheme of the obstacle avoidance part of the architecture is shown in Figure 2.3a, while the scheme of the target acquisition architecture is shown in Figure 2.3b.

For obstacle avoidance, we configured two neuronal populations of 16 neu-



(a) Obstacle avoidance (b) Target acquisition

Figure 2.3: The implemented neuronal architectures for obstacle avoidance and target acquisition. Violet OL and OR node are obstacle detecting neuronal populations. Orange DL and DR are motor driving populations. Violet line of nodes shows a DNF population which represent targets. Thin arrows show excitatory non-plastic connections realised on the ROLLS chip, whereas colors and numbers show the weights (the exact value of the weights is set by the biases, listed in Table 7.1 in the Appendix (Section 7.1.2)). On the chip, both architectures are realised at the same time.

rons each to represent a sensed obstacle to the right ("obstacle right", or OR) and to the left ("obstacle left", or OL) from the robot's heading direction. Each neuron in the OL and OR populations receives a spike for each DVS pixel that

produces an event in the left (right) part of the sensor, respectively (we used the lower half of the sensor for obstacle avoidance). The spiking neurons in the two obstacle populations sum up the camera events according to their neuronal integrate-and-fire dynamics (equations can be found in Appendix). If enough events arrive from the same neighborhood, the respective neuron will fire, otherwise it will ignore events that are caused by the sensor noise. Thus, the obstacle representing neuronal populations achieve basic filtering of the DVS events. The output spikes of the neuronal populations signal the detection of an object in the respective half of the field of view. Each of the obstacle detecting neuronal populations is connected to a motor population "drive left, DL" or "drive right, DR" (with 16 neurons per population). Consequently, if an obstacle is detected on the right, the drive left population is stimulated, and vice versa. The drive populations inhibit each other, implementing a winner-take-all dynamics. Thus, a decision about the direction of an obstacle-avoiding movement is taken and stabilised at this stage by the dynamics of neuronal populations on the chip.
The *drive* populations, in their turn, inhibit both obstacle detecting populations, since during a turning movement of the robot, many more events are generated by the DVS, compared to those generated during translational motion. This inhibition compensates for this expected increase in the input rate, similar to the motor re-afferent signals in biological neural systems (Dean *et al.,* 2009). This modification of the simple Braitenberg vehicle principle is required to enable robust and fast behaviour. The speed of the robot is controlled by a neuronal population, "speed, sp", which receives input from a constantly firing "exc", excitatory population. The latter group of neurons has strong recurrent connections and continually fires when triggered by a transient activity pulse. In an obstacle-free environment, the speed population sets a constant speed for the robot. The obstacle detecting populations OL and OR inhibit the speed population, making the robot slow down if obstacles are present. The decreasing speed ensures a collision-free avoidance manoeuvre. These six populations comprise only 96 neurons, and represent all that is needed to implement the obstacle avoidance dynamics in this architecture (Figure 2.3a). The control signals sent to the robot are, first, the angular velocity that is proportional to the number of spikes per neuron emitted by the two drive populations (eq. 2.1), and, second, the forward velocity, calculated based on the number of spikes per neuron emitted by the

speed population (eq. 2.2):

$$v_a = c_{vel}\left(\frac{\mathrm{N}_{\mathrm{DL}}^{spike}}{\mathrm{N}_{\mathrm{DL}}^{n}} - \frac{\mathrm{N}_{\mathrm{DR}}^{spike}}{\mathrm{N}_{\mathrm{DR}}^{n}}\right), \tag{2.1}$$

$$s = c_{speed}\frac{\mathrm{N}_{sp}^{spike}}{\mathrm{N}_{sp}^{n}}, \tag{2.2}$$

where $\mathrm{N}_{\mathrm{XX}}^{spike}$ are the numbers of spikes, obtained from the respective populations (drive left (DL), drive right (DR), and speed (sp)) in a fixed time-window, we used 500ms and 50ms in an improved version); $\mathrm{N}_{\mathrm{XX}}^{n}$ is the number of neurons in the respective population; and c_{turn} and c_{speed} are turn- and speed-factors (user-defined constants), respectively. Thus, we used neural population dynamics to represent angular and translational velocities of the robot and used the firing rate of the respective populations of neurons as the control variable.

Dynamic neural field for target representation

To represent targets of the navigation dynamics, we use a Dynamic Neural Fields (DNFs) architecture as defined in (Bicho, Mallet, *et al.,* 2000). DNFs are population-based models of dynamics of large homogeneous neuronal populations, which have been successfully used in modeling elementary cognitive function in humans (Schöner & Spencer, 2015), as well as in implementing cognitive representations for robots (Erlhagen & Bicho, 2006; Bicho, Erlhagen, *et al.,* 2011; Sandamirskaya *et al.,* 2013). DNFs can be easily realised in neuromorphic hardware by setting a winner-takes-all (WTA) connectivity network in a neural population (Sandamirskaya, 2014). Each neuron in a soft WTA network has a positive recurrent connection to itself and to its 2-4 nearest neighbors, implementing the lateral excitation of the DNF interaction kernel. Furthermore, all neurons have inhibitory connections to the rest of the WTA network, implementing the global inhibition of a DNF. These inhibitory connections can be either direct, as used here, or be relayed through an inhibitory population, which is a more biologically plausible structure.

In our architecture, we select 128 neurons on the ROLLS chip to represent visually perceived targets. Each neuron in this population receives events from the upper half of each column of the 128x128 sensor frame from the eDVS and integrates these events according to its neuronal dynamics: only events that consistently are emitted from the same column lead to firing of the neuron. The nearby neurons support each other's activation, while inhibiting further neurons

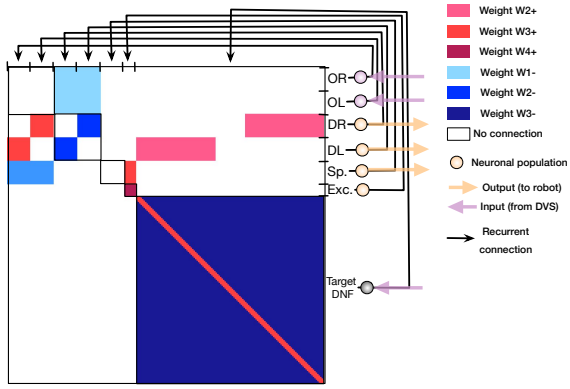


Figure 2.4: The synaptic connectivity matrix, configured on the ROLLS chip to implement the obstacle avoidance and target acquisition architectures. Colors encode different synaptic weights (red for positive and blue for negative connection weights) of the recurrent connections on the chip.

in the WTA population (Figure 2.3b). This connectivity stabilises localised blobs of most salient sensory events, filtering out sensor noise and objects that are too large (inhibition starts to play role within object representation) or too small (not enough lateral excitation is engaged). Thus, the WTA connectivity stabilises the target representation. The target in our experiments was a blinking LED of the second robot, which was detected in the DNF realised on the ROLLS. While this target could be easily detected since the blinking LED produces many events, more sophisticated vision algorithms are being developed to pursue an arbitrary target (Moeys *et al.,* 2016). The target population was divided in three regions: neurons of the DNF that receive inputs to the left from midline of the DVS frame drive the “drive left” population, whereas neurons that receive input from the right half of the DVS frame drive the “drive right” population. We did not connect the central 16 neurons of the target DNF to the drive populations to ensure more smooth target pursue when the target is in the center of the DVS frame (Figure 2.3b).

Combining obstacle avoidance and target acquisition

The two neuronal populations that ultimately determine the robot’s steering direction (DR and DL) sum-up contributions from the obstacle-representing populations and the target-representing WTA population (Figure 2.3). The obstacle contribution is made effectively stronger than the target contribution by

setting the ROLLS biases accordingly. Thus, in the presence of an obstacle in the robot's field of view, an obstacle avoidance manoeuvre is preferred. Figure 2.4 shows the connectivity matrix used to configure the non-plastic connections on the ROLLS chip to realise both obstacle avoidance and target acquisition. This plot shows the weights of non-plastic synapses on the ROLLS chip (blue being the negative weights and red the positive weights; the same color code is used for the different weights as in Figure 2.3), which connect groups of neurons (different populations, labeled on the right side of the figure) among each other. Within-group connections are marked with black squared frames on the diagonal of the connectivity matrix. Violet and orange arrows show inputs and outputs of the architecture, respectively. This connectivity matrix is sent to the ROLLS device to configure the neuronal architecture on the chip, i.e. to "program" the device.

2.3 Demonstrations

We verified the performance of our system in a number of demonstrations, reported next. Overall, over 100 runs were performed with different parameter settings. In the following, we will provide an overview for the experiments and describe a few of them in greater detail to provide intuition of how the neural architecture works. For most experiments, we let the robot drive in a robotic arena with a white background and salient obstacles. We used a tape with a contrastive texture to make the walls of the arena visible to the robot. In four runs, we let the robot drive for several minutes freely in the office.

2.3.1 Probing the obstacle avoidance: a single static obstacle

In the first set of experiments, we let the robot drive straight towards a single object (a colored block 2.5cm wide and 10cm high) and measured the distance from the object at which the robot crossed a virtual line perpendicular to the robot's initial heading direction, on which the object is located (e.g., see the distance between the robot and the 'cup' object at the last position of the robot in Figure 2.5). We varied the speed factor of the architecture from 0.1 ($\sim$0.07m/s) to 3.0 ($\sim$1m/s) and have verified the effectiveness of the obstacle avoidance manoeuvre. Furthermore, we have increased the turning factor from 0.5 to 1.0 to improve performance at high speeds and have tested color-dependence of the obstacle perception with the DVS. Table 2.1 shows results of these measurements.

Table 2.1: Collision avoidance at different speeds: distance to the obstacle when crossing the obstacle-line (mean over 3 trials ± standard deviation in [cm]) at different speed- and turn-factors and for different colors of the obstacle. * signifies trials when a collision happened.

Speed/Turn	0. 1/0.5	0.5/0.5	1/0.5	1/1	1.5/1	2/1	3/1
Blue	7.0±1.0	10.3±0.6	7.7±1.5	19.3±2.1	16.3±3.3	10.8±2.6	0*
Red	0*	2.3±0.6	4.7±0.6	10.7±1.2	9.7±3.5	5.0±1.0	0*
Yellow	0*	0*	0*	7.0*±6.1	0*	0*	0*

Each trial was repeated 3 times and mean over the trials was calculated. The table allows to note the following characteristics of the architecture at the chosen parametrisation. First, the performance drops at very low speeds (speed factor 0.1), especially for red and yellow objects, due to an insufficient number of DVS events to drive the neuronal populations on ROLLS. Second, there is a trade-off between this effect and the expected decay in performance (in terms of the decreasing distance to the obstacle) with increasing speed. Thus, at a turning factor 0.5, best performance is achieved for the blue object at speed factor 0.5 and for the red object at speed factor 1. Distance to the obstacle can be further increased by increasing the turn factor. Thus, at turn factor 1 and speed factor 1 best performance (i.e. largest distance to the obstacle) can be achieved for both the blue and red objects. Yellow object provides too little contrast to be reliably perceived by the DVS in our set-up. Figure 2.5 demonstrates how the neuronal architecture on the ROLLS chip realises obstacle avoidance with the Pushbot. On the left, an overlay of video frames (recording the top view of the arena) shows the robot's trajectory when avoiding a single obstacle (here, a cup) in one of the runs. Numbers (1-3) mark important moments in time during the turning movement. On the right, summed activity of the neuronal populations on the ROLLS device is shown over time. The same moments in time are marked with numbers as in the left figure. In this case, already the obstacle detecting populations had a clear "winner" – the left population forms an increasing activity bump over time, which drives the "drive right" population, inducing a right turn of the robot. The bottom plot shows the commands that are sent to the robot (speed and angular velocity): the robot slows down in front of the obstacle and turns to the right.

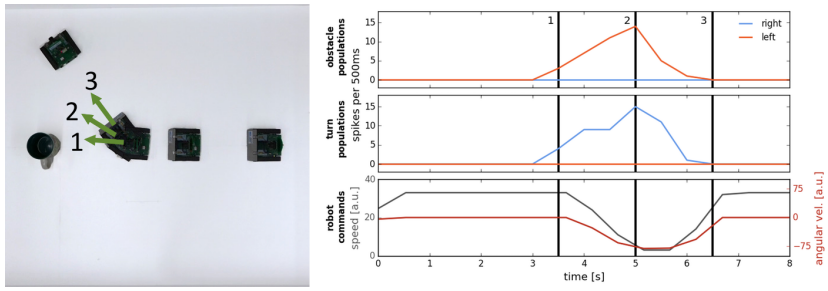


Figure 2.5: An example of an obstacle avoidance manoeuvre. **Left**: Overlay of video frames showing the trajectory of the robot. **Right**: activity of the neuronal populations on the chip (Top: the left and right obstacle detecting populations; Middle: the left and right drive populations), and the motor commands, sent to the robot (Bottom plot).

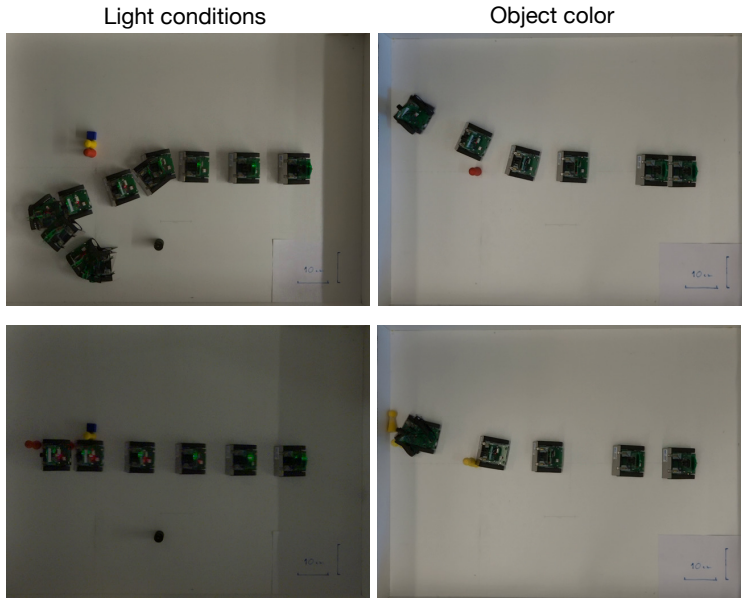


Figure 2.6: Exemplary experiments showing successful (top row) and unsuccessful (bottom row) obstacles avoidance manoeuvres in different light conditions (left) and with obstacles of different colors (right).

2.3.2 Avoiding a pair of obstacles

We repeated the controlled obstacle avoidance experiment with two and three blocks in different positions. Each configuration was tested twenty times without crashes at speed 0.35m/s (speed factor 0.5). Figure 2.7 shows an exemplary run that explains *how* the robot avoids a pair of obstacles. This example is important, since in the attractor dynamics approach to navigation, distance between the two objects determines a decision to move around or between the objects. Snapshots from the overhead camera are shown on the left of Figure 2.7. Output of the DVS, accumulated in 500*ms* time windows around the time when the snapshots were taken[3], is shown in the second column, and the spiking activity of neuronal populations recorded from the ROLLS chip is shown in the two right-most columns. Activity is shown of the obstacle representing left (red) and right (blue) neuronal populations (third column), the left (red) and right (blue) drive populations, and the speed population (gray, forth column). Each of these populations has 16 neurons, dots represent their spikes[4].

At the moment, depicted in the top row of Figure 2.7, the robot senses an obstacle on the right, although the DVS output is rather weak. Note that the neuronal population filters out sensory noise of the DVS and only detects events that cluster in time and in space. The robot turns left, driven by the activated *drive left* population and now the obstacle on the right becomes visible, providing a strong signal to the *right obstacle* population and, consequently, to the *drive left* population (second and third row). Eventually, the obstacle on the right dominates and the robot drives past both obstacles on the left side (forth row). Thus, with the chosen parametrisation of the neuronal network architecture, the robot tends to go around a pair of objects, avoiding the space between them. This behaviour could be changed, making the connections between the obstacle representing populations and drive populations stronger. However, for a robot equipped with a DVS, such strategy is safer, since for homogeneous objects, the DVS can only sense the edges, where a temporal contrast change can be induced by the robot's motion. The robot thus might miss the central part of an object and avoiding pairs of close objects is a safer strategy. Adaptive connectivity that depends on the robot speed is also feasible.

[3] We dropped 80% of DVS events randomly in our architecture; moreover, we only used 5% of all remaining events for plotting.

[4] Only 5% of the ROLLS spikes (every 20^{th} spike) are shown in all our plots.

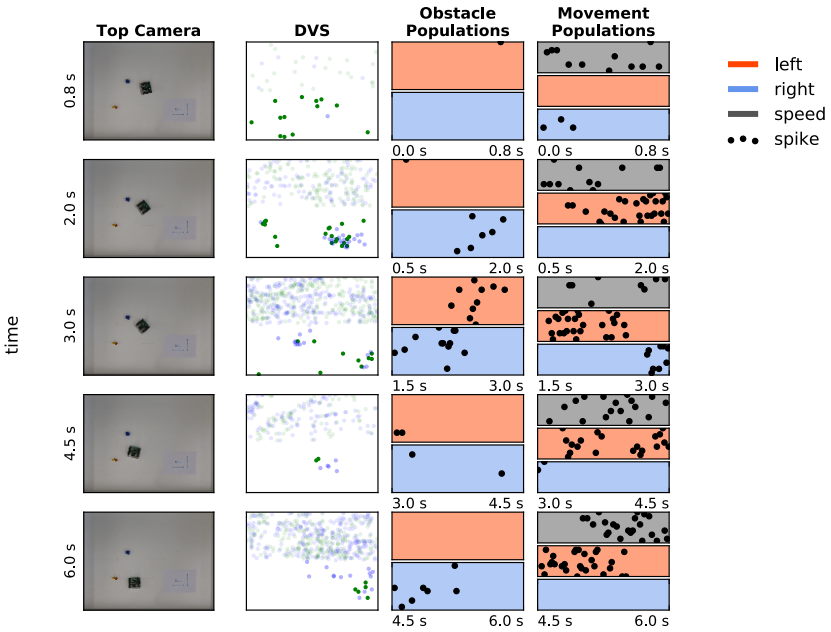


Figure 2.7: Avoiding a pair of obstacles. **First column**: Snapshots of four moments in time during avoidance of a cup, moved into the robot's trajectory. **Second column**: DVS "frames" – events, accumulated over a 0.5s time window. Green dots are off events, blue dots are on events. Events in the upper part of the frame were not considered for the obstacle avoidance. **Third column**: Activity of the obstacle representing populations in 0.6-1.5 seconds before the camera snapshot in the first column was taken (red – left population (n_{OL}), blue – right population (n_{OR}); each population has 16 neurons). **Forth column**: Activity of the drive left (red), drive right (blue), and speed population on the ROLLS chip in the same time as on the plots in column 3.

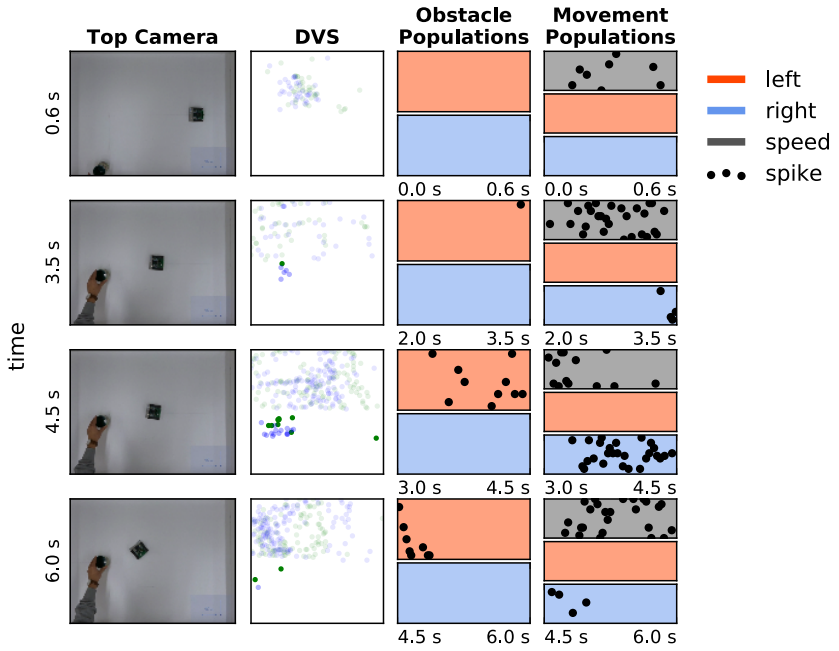


Figure 2.8: Avoiding a moving obstacle. The same arrangement is used as in Figure 2.7. See main text for the discussion.

2.3.3 Avoiding a moving obstacle

In these experiment, the robot is driving straight in the arena while we move an obstacle (a coffee mug) into its path. We repeat this experiment six times with varying speed factors (0.1-2) of the robot. The robot was capable to avoid collisions in all tested cases. In fact, avoiding a moving obstacle is more robust than avoiding a static obstacle because the moving obstacle produces more DVS events than a static one at the same robot speed. Figure 2.8 shows how the robot avoids a moving obstacle. The same arrangement of plots was used as in Figure 2.7, described in Section 2.3.2. The robot was moving with $c_{speed} = 0.5$ (0.35m/s) here, the cup was moved at approx. $0.20m/s$.

2.3.4 Cluttered environment

In the following set of experiments, we randomly placed obstacles (8-12 wooden pieces) in the arena and let the robot drive around at an average speed (0.35m/s). We analysed the performance of the architecture, suggesting a number of modifications to cope with its limitations. Figure 2.9 demonstrates behaviour of

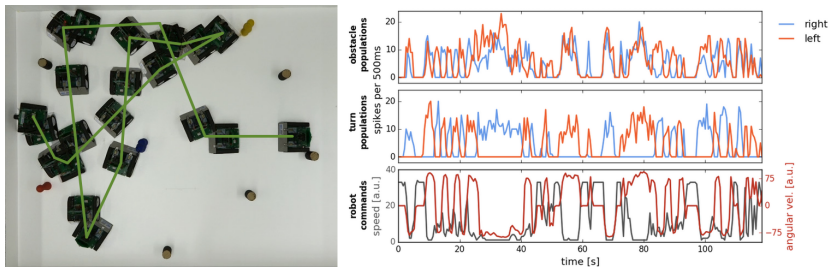


Figure 2.9: Navigation in a cluttered arena. **Left**: Overlayed frames from the video, recoding the robotic arena from the top. Green line markes the path of the robot. **Right**: Summed activation of neurons in populations on the ROLLS chip over the time of the experiment. Obstacle and turn (left and right) population are shown, as well as the commands sent to the robot (angular velocity and speed).

the obstacle avoidance system in a cluttered environment. In particular, we let the robot drive in an arena, in which 8 obstacles were randomly distributed. The robot successfully avoids obstacles in its way with two exceptions: The robot touches the blue obstacle in the center of the arena, which entered the field of view too late for a manoeuvre, and also collides with the yellow object, which did not provide enough contrast to produce the required number of DVS events. These collisions point to two limitations of the current setup, which, first, uses single camera with a narrow field of view and, second, drops 80% of events to improve signal to noise ratio (the latter deprives performance for objects with low contrast against the background). Using more sophisticated noise filter would improve visibility of the faint obstacles. Note that we used rather small objects on these trials (blocks of 2x5cm), which posed a challenge for the event-based detection, especially taking into account our very simplistic noise-reduction strategy.

To improve behaviour in a cluttered environment, we modified the architecture, adding two more populations on the ROLLS chip, which receive input from the inertia measurement unit of the Pushbot and which suppress obstacle populations when the robot is turning. Moreover, we replaced the homogeneous connections between the obstacle and the drive populations with graded connections that become stronger for obstacles detected in the center than in the periphery of DVS field of view. This allows the robot to make shorter avoidance manoeuvres and avoid obstacles in a denser configuration at a higher speed. Figure 2.10 shows a successful run with the modified architecture. Here, we also changed the sampling mechanism used to calculate the robot commands,

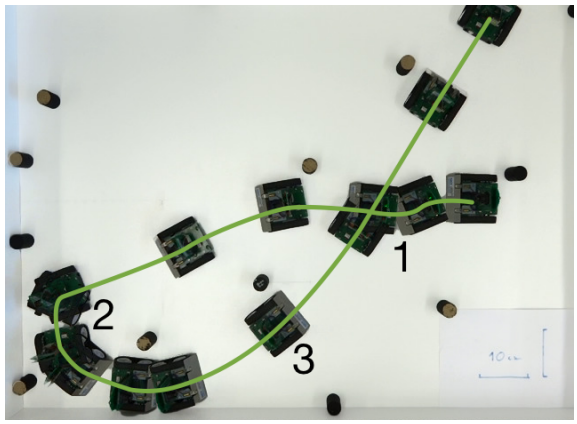


Figure 2.10: Successful run in a cluttered environment with a modified neuronal architecture. Overlay of the overhead-camera frames.

replacing a fixed time window with a running average. This allowed us to avoid obstacles in the cluttered environment without collisions at speed as high as 0.5m/s.

2.3.5 Variability of behaviour

Since behaviour of our robot is controlled by activity of neuronal populations, implemented in analog neuromorphic hardware, the behaviour of the robot has some variability, even when exactly the same parameters of the architecture and the same hardware biases are used. Despite this variability, the robot's goal – avoiding obstacles – remains fulfilled. Such variability of behaviour can be used as a drive for exploration, which may be exploited in learning scenarios in more complex architectures, built on top of our elementary obstacle avoidance system. Figure 2.11 demonstrates variability behaviour of our neuronal controller. In the figure, we show three trials, in which the robot avoids a two-blocks configuration, starting from exactly the same position and with the same configuration of the neuronal controller (speed factor 0.5, turn factor 0.5). Mismatch in the neuronal populations implemented in analog neuromorphic hardware, variability of the DVS output, and its dependence on the robot's movements lead to strong differences in trajectories. In particular, in the case shown in Figure 2.11, the trajectories may bifurcate and the robot might avoid the two obstacles on the right, or on the left side.

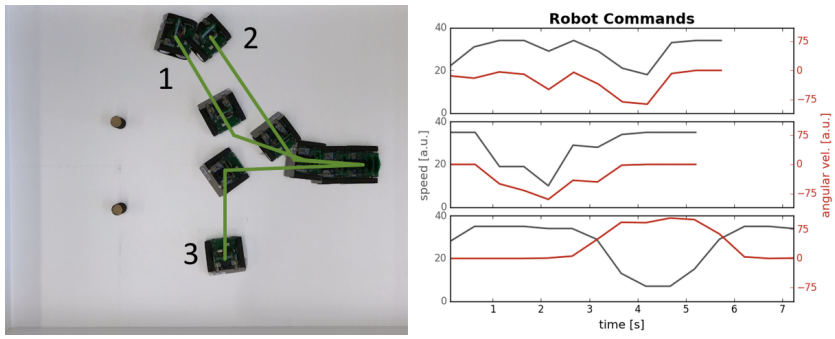


Figure 2.11: Variability of the robot's behaviour. **Left**: Overlay of video camera frames recording the robot, avoiding a pair of obstacles; top view. Three different trials are recorded and overlayed here (trajectories are shown with green lines 1-3). **Right**: Velocity commands, received by the robot from the neuronal architecture (angular velocity and speed) for the three trials (from top to bottom).

2.3.6 Obstacle-avoidance in a real-world environment

Finally, we tried our architecture outside of the arena as well. The robot was placed on the floor in the office and drove around avoiding both furniture and people. The high amount of background activity compared to the arena did not diminish the effectiveness of the architecture: in four 0.5-1.5-minutes long trials, the robot only crashes once after it manoeuvred itself into a dark corner under a table where the DVS sensor could not provide sufficient information to recognise obstacles. Figure 2.12 shows an example of the Pushbot robot driving in the office environment. On the left, three snapshots from the video camera recording the driving robot are shown (full videos can be see in the Supplementary material). The snapshots show the robot navigating the office environment with its task being to avoid collisions. The middle column of plots shows pairs of eDVS events, accumulated over 500 *ms* around the moment in time in the corresponding snapshot on the left, and respective histograms of events from the center region, used for obstacle avoidance. Events above the mid-line of the eDVS field-of-view are shown with transparency to emphasise that they were not used for obstacle avoidance: only events from the region of the eDVS field-of-view between the two vertical lines in Figure 2.12 were used. Histograms below the eDVS plots show the events from this region of the field of view, summed over the eDVS columns. These events drive the obstacle left (red colored part of the histogram) and obstacle right (blue part of the

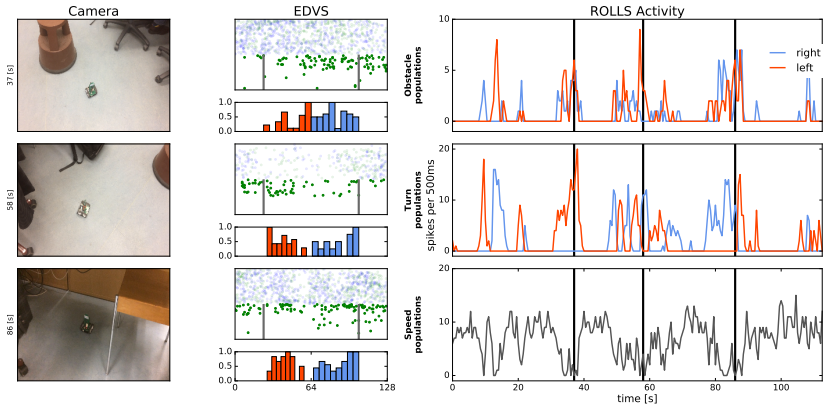


Figure 2.12: Robot driving in the office environment. **Left**: Snapshots from the video camera showing robot at three time points during the experiment. **Middle**: Events from the DVS camera and histogram of these events, binned over 500 *ms* in columns in the region between two vertical lines, which were used to drive obstacle populations on the ROLLS. Each pair of the eDVS events and histogram corresponds to the time point of the video frame in the **Left** column. Note that 80% of events are randomly dropped here and only "on" events are shown in the relevant region (lower part of the screen). Events above the midline of the image sensor are shown with transparency (these events were not used for obstacle avoidance). **Right**: Activity of the obstacle (left and right), drive (left and right), and speed neuronal populations over time (summed activity across each population). Vertical lines mark time point that correspond to the video frames in the **Left** column.

histogram) neuronal populations on the ROLLS chip. The right column shows activity of the neuronal populations on the ROLLS chip over time, as in the previous chapters/papers/Figures. Black vertical lines mark time moments that correspond to the three snapshots in the left column. These plots allow to see that although the left and right obstacle populations are often activated concurrently, only one of the drive populations (either left or right) is active at any moment, leading to a clear decision to turn in either direction in the presence of perceived obstacles. The speed plot shows that movement of the robot is not very smooth – it slows down and accelerates often based on the sensed presence of obstacles. This behaviour is improved in the modified architecture, briefly described in Section 2.3.4.

When driving around the office, robot faced very different lighting conditions, as can be seen already in the three snapshots presented here. This variation in lighting conditions did not effect obstacle avoidance in most cases, since the DVS is sensitive to relative change of each pixel's intensity, which varies less than the absolute intensity when the amount of ambient light changes. However, in an extreme case, shown in the lower snapshot in Figure 2.12, the robot collided with the metal foot of the chair. This was the only collision recorded.

2.3.7 Target acquisition

In addition to obstacle avoidance we also tested target acquisition in ten experiments using a second robot with a blinking LED as target. The robot successfully turns and drives towards the target every time (at speed and turn factors =0.5). In 8 out of 10 experiments the target is recognised as an obstacle when approached and is avoided; in two experiments, the robot failed to recognise target as obstacle after approaching it. Obviously, the simple visual preprocessing that we used did not allow us to distinguish the target from obstacles (other than through their position in the upper or lower part of the field-of-view of the DVS). Moreover, we would need an object detection algorithm to detect the target and segregate it from the background. This vision processing is outside the scope of our work, but there is a multitude of studies going in this direction (Moeys *et al.*, 2016) using modern deep/convolutional neural networks learning techniques.

Figure 2.13 shows target acquisition for a static target and demonstrates that the robot can approach the target object. At a short distance, the obstacle component takes over and the robots turns away after approaching the target. The figure shows the overlayed snapshots from the overhead camera, showing how

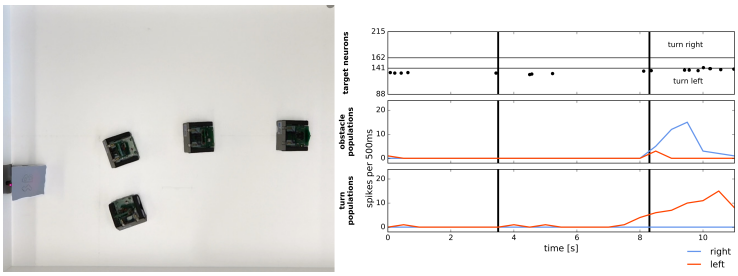


Figure 2.13: Simple target acquisition: single stationary target. **Left**: Overlay of video frames from the overhead camera. The robot approaches a stationary target on the left-hand side of the arena from right to left. The robot turns left toward the target until it perceives it as an obstacle and makes an obstacle avoidance manoeuvre. **Right**: Time-course of the spiking activity (raster plot) of the target-representing (WTA) neurons on the ROLLS chip (top plot) and summed (over 500ms and over populations) activity of neurons in obstacle representing and drive populations on the ROLLS chip. Vertical lines mark time points that correspond to two middle positions of the navigating robot.

the robot turns toward the second robot, standing on the left side of the image. When getting close to the second robot (approx. 10cm), the robot perceives the target as an obstacle, which has a stronger contribution to its movement dynamics and the robot turns away. On the left, the spiking activity of the target representation on the ROLLS chip is shown (raster plot where each dot represents a spike[5]). We can see that the robot perceives its target consistently on the left. After the eighth second, the obstacle contribution on the right becomes dominant and the robot turns left strongly.

Figure 2.14 shows how the robot can chase a moving target. We have controlled the second Pushbot remotely and have turned its LED on (at 200 Hz, 75% on-time). The LED provided a rather strong (though spatially very small) input to the DVS of the second, autonomously navigating robot. This input was integrated by our target WTA (DNF) population, which, however, also received a large amount of input from the background (in the upper part of the field of view the robot could see behind the arena's walls). Input from the localised LED was stronger and more concise than more distributed input from the background and such localised input was enhanced by the DNF's (WTA's) lateral connections. Consequently, the respective location in the target WTA formed a "winner" (localised activity bump in the DNF terminology) and inhibited the interfering inputs from other locations. In the figure, four snapshots of the video recording

[5]Remember, that only 5% (every 20th) of all spikes from the ROLLS processor are shown.

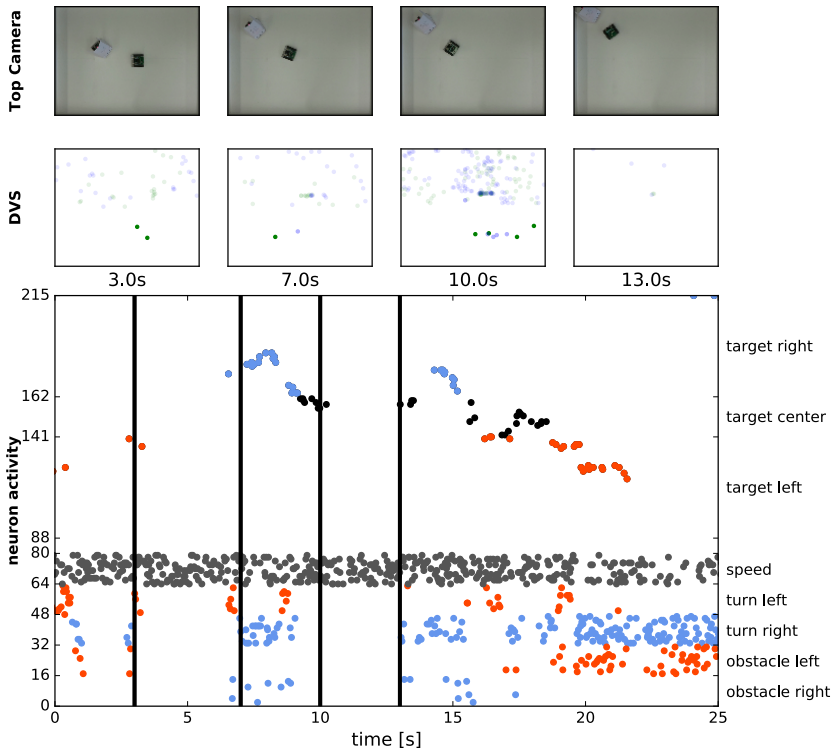


Figure 2.14: Chasing a moving target. **First row**: Snapshots from the overhead camera showing the robot controlled by the ROLLS chip chasing a manually controlled robot. **Second row**: Summed eDVS events from 500ms time windows around the same moments. Events from the upper part were used for target acquisition, events from the lower part – for obstacle avoidance. **Bottom row**: Spiking activity of all neurons on the ROLLS chip over the time of the experiment. Vertical line show the moments in time, selected for the first two rows. Red dots are spikes from the "left" populations and blue dots are spikes from the "right" populations.

the two robots are shown (top row). The leading robot was covered with white paper to reduce interference from the obstacle avoidance dynamics as the robots get close to each other (the space in the arena and the small size of the blinking LED forced us to put the robots rather close to each other, so that the target robot could be occasionally perceived as an obstacle). In the second row in Figure 2.14, the summed over 500ms events of the DVS are shown, around the same time points as the snapshots. Only the upper part of the field-of-view was considered for target acquisition. This part is very noisy, since the robot "sees" outside the arena and perceives objects in the background, which made target acquisition very challenging. Still, the blinking LED provided the strongest input and in most cases the target DNF was able to select its input as the target and suppress the competing inputs from the background – see activity of neurons in the target DNF in the bottom plot. This last plot shows spiking activity of 215 neurons of the ROLLS chip, used to drive the robot (we don't show the constantly firing n_{exc} population here). We can see that the target DNF (WTA) successfully selects the correct target in most cases, only loosing it from sight twice, as the robot receives particularly many DVS events from the background during turning. The lower part of this raster plot shows activity of the obstacle populations, the drive populations, and the speed population, thus the dynamics of the whole architecture can be seen here.

2.4 Discussion

This paper presents a neuronal architecture for reactive obstacle avoidance and target acquisition, implemented using a mixed-signal analog/digital neuromorphic processor (Qiao *et al.,* 2015) and a silicon retina camera DVS as the only source of information about the environment. We have demonstrated that the robot, controlled by interconnected populations of artificial spiking neurons, is capable of avoiding multiple objects (including moving objects) at an average movement speed (up to 0.35m/s with our proof of concept setup). We have also demonstrated that the system works in a real-world office environment, where background clutter poses a challenge for the DVS on a moving vehicle, creating many distracting events. We demonstrated that also the target acquisition neural architecture can cope well with this challenge, which was relevant even in the robotic arena. The distributed DNF representation of the target, supported by lateral interactions of the WTA neuronal population, enabled robust detection and reliable selection of the target against background. The reactive approach

to obstacle avoidance that we adopt in this work has a long history of success, starting with the neurally inspired turtle robot more than half a century ago, as reviewed by Holland, 1997. Later, Valentino Braitenberg analysed a number of hypothetical vehicles, or creatures, that use reactive control to produce complex behaviours (Braitenberg, 1986). His controllers were realised as simple "nervous systems" that directly linked the sensors to the motors of the vehicle. Using similar sensorimotor, or behavioural modules as building blocks, Rodney Brooks developed a behaviour-based controller paradigm for roaring vehicles, known as "subsumption architecture" (Brooks, 1991). Although this framework did not scale well for complex tasks and is not ideally suited for online learning methods, this type of controller is at the heart of highly successful real-world robotic systems such as the autonomous vacuum cleaners, and has been adopted, to some extent, in a wide range of impressive controllers for autonomous robots (e.g., Khansari-Zadeh & Billard, 2012). The dynamical systems approach to robot navigation (Schöner *et al.,* 1995) is an attempt to mathematically formalise reactive control for autonomous robots using differential equations that specify attractors and repellors for behavioural variables that control the robot's heading direction and speed (Bicho, Mallet, *et al.,* 2000). In this framework, obstacle avoidance has been integrated with target acquisition and successful navigation in an unknown environment has been demonstrated both for vehicles and robotic arms (Reimann *et al.,* 2011). This approach is similar to another successful reactive approach to obstacle avoidance: the potential field approach (e.g., Haddad *et al.,* 1998), in which the target creates a global minimum in a potential that drives the robot, whereas obstacles create elevations in this potential. However, the use of Cartesian space instead of robot-centered velocity space used in this potential field approach makes it prone to getting trapped in local minima.

In mixed-signal analog /digital neuromorphic hardware, the neuronal dynamics is taken care of by the physics of analog electronic circuits, avoiding loosing digital computational resources on simulating them. Thus, neuromorphic implementation of simple biologically inspired obstacle-avoidance architectures can lead to low-latency (on the order of microseconds) and power-efficient (on the order of milliwatts) solutions, analogous to the ones used by insects. In contrast, more conventional obstacle-avoidance systems require a substantial amount of computing resources to process and store sensory data, detect obstacles, and compute motor commands. Neuromorphic implementation of such low-level processing will allow to use analogue sensory signals directly, avoiding their digital representation and storage, while at the same time allowing to build complex

neural-network based computing architectures, that could be used for solving cognitive tasks, such as task planning, map building, or object recognition. We consider the work proposed as a first feasibility study, which still has a number of limitations that we will address in our future work. The main limitation is variability of neuronal behaviour because of parameter drift on the analog hardware: The parameters of the hardware neural network change the network properties as the experimental setup conditions (temperature, humidity, etc.) change. This is a serious limitation of the hardware used, which makes in challenging to implement complex architectures that have to balance contributions of different behavioural modules (e.g. controlling turning and forward velocities, or obstacle avoidance and target acquisition). We are currently working on algorithms and methods for automatically re-tuning these parameters in a principled fashion with optimisation and machine learning techniques. In addition, we are designing new versions of the neuromorphic hardware with on-board stabilisation of the chip parameters, and more resources for simplifying the fine-tuning process of the architectures. However, approach employed here – use of populations of artificial neurons in place of single nodes in the architecture – allowed us to generate behaviour with the state of the art analogue neuromorphic hardware. Apart from the hardware limitations, our simple architecture currently allows robust obstacle avoidance at moderate speeds (approximately 0.35 m/s). Since the robot slows down when an obstacle is detected, movement appears to be "jerky". Although the smoothness of the robot movement could be improved by tuning the coupling strength between the obstacle and drive populations, the best solution would involve improving the visual pre-processing stages. In our setup, the DVS detects local contrast changes and produces different amount of events depending on the objects in the environment, but also modulated by the robot translational and rotational movements. Currently we ignore about 80% of all DVS events to remove both noise and to reduce bandwidth. This very basic strategy improves the signal to noise ratio, because the architecture enhances the spatially and temporally coherent inputs and suppresses the effect of random inputs. However, we plan to study a more principled approach to pre-processing and noise reduction, and to investigate other biologically inspired architectures for obstacle avoidance, for example inspired by the fly's Elementary Motion Detector (EMD) (Hassentstein & Reichardt, 1956) or the locust's Lobula Giant Movement Detector (LGMD) (looming detector) (Gabbiani *et al.,* 2002; Rind & Santer, 2004). We are currently working on neuromorphic implementation of these algorithms (Milde, Sandamirskaya, *et al.,* 2016; Salt *et al.,*

2017). Moreover, the 500*ms* time window that we used to create plots of DVS events and average spiking activity was also used in our controller for counting spikes when calculating motor commands, sent to the robot. In our preliminary experiments on optimising the controller, we have reduced this time window to 50*ms* and, more importantly, replaced it with a sliding-window calculation of the average firing rate of the drive and speed neuronal populations. A more principled solution to this problem would be development of a more direct hardware interface between the spiking neuromorphic processor and the robot's motors, so that spikes can control the motor rotation directly, as suggested by Perez-Peña, Morgado-Estevez, *et al.,* 2013.

Our target acquisition network can also be further improved: The main strategy will be to introduce target representations in a reference frame that moves with the robot, but has a fixed orientation. Such representation will allow the robot to turn back to a target that has been lost from sight due to an obstacle avoidance manoeuvre. Furthermore, increasing the strength of lateral interactions in the WTA (DNF) population will allow to stabilise the target representation, allowing it to form a "working memory", which will support target acquisition behaviour in cluttered environments. To still make the system reactive and allow it to follow the visible target, control of the strength of lateral interactions will be introduced, increasing their strength when target is being lost from view and decreasing their strength when the target is visible. Detecting the target based on its features perceived with a DVS is a separate topic of ongoing research both in our lab and worldwide (e.g., Lagorce, Ieng, *et al.,* 2015).

Despite of this list of necessary improvements, our neuromorphic architecture is an important stepping stone towards robotic controllers, realised directly in neurally inspired hardware, being the first architecture for closed-loop robot navigation that uses analog neuromorphic processor and minimal preprocessing of visual input, obtained with a silicon retina DVS. Such neuromorphic controllers may become an energy efficient, fast, and adaptive alternative to conventional digital computers and microcontrollers used today to control both low-level and cognitive behaviours of robots. While *neural network* implementations using the conventional computing architecture are typically time- and energy consuming, implementation of neuronal architecture using analog neuromorphic hardware approaches the efficiency of biological neural networks. Building neuronal models for higher cognitive function using, for instance, the framework of Dynamic Neural Fields (Sandamirskaya, 2014) or the Neuro-Engineering Framework (Eliasmith, 2005), will allow to add more complex behaviours to the robot's

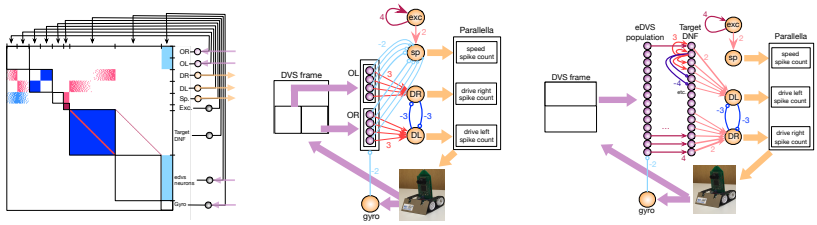
repertoire, e.g. finding a particular object, grasping and transporting it, as well as map formation and goal-directed navigation, which is the goal of our current research efforts.

Postamble

We showed that despite the presence of approximately 20% noise events emitted by the DVS, as well as the inherent noise due to mismatch of mixed-signal analogue/digital neuromorphic processors, the fully neuromorphic sensory-processing system is capable of generating robust avoidance behaviour in a variety of environments, lighting conditions and velocities. We used populations of neurons, which coded for the same information, e.g. "turn left", to average out the variability introduced by device mismatch. This clearly is only compensating the mismatch, not actually taking advantage of the device variability. In the third part of this thesis, we will address the issue of how we can take advantage of device mismatch and that it represents a crucial feature for stable computation of SNNs emulated on neuromorphic processors.

The low-pass property of synapses which connect two neurons is capable of smoothing out the high-frequency components of the noise introduced by the sensory system. While the neuromorphic processor used in this study only comprises 256 neurons, which share the same biases, the weight resolution was extremely low with only two bits. To account for this rather hard constraint, we exploited the fact that we can increase the number of synaptic connections between any two populations. By doing so we not only could increase the weight resolution, but we also could guarantee smooth behavioural decisions as the retinal position of an obstacle was decisive for the turning amplitude. In other words, neuronal populations with a centrally located receptive field had more synaptic connections to their respective drive population compared to neuronal populations with a peripherally located receptive field (see Figure 2.15 and Section 2.3.4). However, as soon as the system encountered an obstacle and performed an avoidance manoeuvre, the rotational ego-motion produced a flood of events across the entire sensor array, which are not informative as the decision to turn was already made. To cope with the flood of events, we installed a saccadic suppression mechanism, which prevents events from entering the system as long as the system is turning. We used information about the angular velocity of the agent as provided by a gyroscope to successfully inhibit incoming events (see Figure 2.15). To shut down all activity seems rather crude, however, it turned out to be good enough in the context of this very simple task. We will address the issue of how to predict sensory and neuronal activity in the third part of this thesis.

This approach to scene understanding is rather rudimentary. The system clearly



(a) (b) (c)

Figure 2.15: The modified network architecture used in in the cluttered environment experiments (see Section 2.3.4) a) The network architecture expressed as a connectivity matrix set on the ROLLS device. Red are excitatory weights, blue - the inhibitory ones. Note the gradient in connectivity to compensate for limited weight resolution. b) Obstacle avoidance part of the neural architecture. c) Target acquisition part of the neural architecture. In both applications the information about the angular velocity of the agent was used to suppress incoming events during turning manoeuvres.

is of a reactive nature only and does not incorporate any further information about the environment beyond the presence of moving edges at given locations in the retina. Some studies suggest that the number of events is proportional to the amplitude of optic flow (Censi & Scaramuzza, 2014, for definition see Section 3.1 & 3.1.1). This can be easily understood as close-by objects occupy larger retinal area, thus producing more events, than distant ones. However, the encoded amplitude of optic flow is transmitted in the rate of activity of pixels in a spatial neighbourhood and, as it is encoded in a rate, is only meaningful if averaged over some period of time. This not only constrains the spatial resolution, but more importantly it does not exploit the high temporal resolution as provided by the DVS. In the next chapter, we are going to address the issue of extracting information from the precise relative timing of pairs of events emitted from spatially adjacent pixels. The computed temporal correlation of these events is proportional to the amplitude of optic flow. If this computation is done during translational movement the incorporated information about the strength of optic flow is anti-proportional to the relative distance of a close-by object. Hence, we will demonstrate how to incorporate distance information into the decision making of avoiding collisions with nearby obstacles.

3

A sense of motion and distance

Preamble

In the last chapter we showed that neuromorphic sensory-processing systems are capable of real-time, closed-loop reactive collision avoidance and target acquisition. The developed neural architecture is robust to noise present at the sensory and processing level. The demonstrated behaviour constitutes the most simple form of visual scene understanding, as the system generates appropriate avoidance manoeuvres without extracting any features, but just by low-pass filtering incoming events in large, homogeneously weighted receptive fields.
The architectural and algorithmic solution presented in the previous chapter has two major drawbacks: (i) it is hard-coded, meaning that it is not capable of adjusting its own parameters, i.e. synaptic weights, and (ii) the system only integrates incoming events without extracting any spatio-temporal features. One out of many more relevant spatio-temporal features in the context of autonomous navigation is the relative retinal velocity of an object present in the vicinity of the system, which correlates under some assumptions with the relative distance between the agent and the object. Before we address the first issue in the third part of this thesis, let us focus first on how to incorporate distance information in a network of spiking neurons. As we will see later in more detail, one way to extract distance information while moving through an environment is to estimate optic flow (Gibson, 1950, and see Section 3.1 & 3.1.1). Optic flow has been subject to research in the context of navigation, visual scene segmentation and object tracking for more than 60 years. While most of this body of research developed theoretical frameworks and applications using consecutive frames, more recent investigations developed algorithms to extract optic flow from a

stream of asynchronous events (see Section 3.1.1). The studies discussed below extract optic flow from an event-based vision sensor and encode the information about the amplitude of optic flow as an absolute, analogue voltage or in the output rate-code of a regular spiking neuron (see Section 3.5).

In contrast, in this chapter, we try to formulate a neurally implemented algorithm which additionally to a rate encoding of the amplitude of optic flow also represents this information in a temporal encoding. This additional encoding step will help us to understand the mystery of representing time in ongoing processing and how we can preserve temporal information in downstream processing stages. The following chapter was published as Milde, M. B., Bertrand, O. J. N., Ramachandran, H., Egelhaaf, M. & Chicca, E. Spiking Elementary Motion Detector in Neuromorphic Systems. *Neural computation* **30,** 2384–2417 (2018) [1] and partially adapted for this thesis.

This work presents the smallest, to this date, size mixed-signal analogue/digital solution to estimate event-based optic flow. The developed architectural algorithm features parallel rate- and temporal-encoding of information and attaches meaning to the timing of spikes for downstream processing and decision making stages. The core of the proposed solution is a time-varying adaptation of synaptic efficacy via gain modulation (see Section 3.2.2). This principle not only allows an agent to navigate in outdoors cluttered environments (see Section 3.4), but constitutes a general event-based temporal correlation detection mechanism which might open new perspectives on how to gate information in SNNs based on the relative timing of spikes.

[1] The article is under copyright (Neural Computation 30, 2384–2417 (2018) ©2018 Massachusetts Institute of Technology doi:10.1162/neco_a_01112) and can be accessed via the following link: https://www.mitpressjournals.org/doi/abs/10.1162/neco_a_01112

3.1 Introduction

Both animals and humans move most of the time while interacting with the world. This self-induced motion, i.e. ego-motion, generates continuous change in the retinal image of the animal/human. Whilst the encoding of motion in insects (e.g. flies and bees) is done with graded potential (modelled with the classical Reichardt detector (Hassentstein & Reichardt, 1956; Borst & Helmstaedter, 2015)), dedicated structures in the mammalian brain encode motion information using action potentials, i.e. spikes. However, the precise mechanisms and circuitry to encode motion in cortical structures remains elusive and is subject to ongoing research (Foster *et al.,* 1985; Perrone & Thiele, 2002; Priebe *et al.,* 2006; Rokszin *et al.,* 2010). We know that a dominant mode of communication of information in the vertebrate nervous system are spikes. Hence, elementary motion is expected, in principle, to be encoded with spikes at some processing stage. The classical Elementary Motion Detector (EMD) model (Hassentstein & Reichardt, 1956) does not account for spike-based motion estimation.

First-order motion, i.e. elementary motion, can be described by the characteristic pattern of changes of brightness induced by the motion of objects in the visual scene. This pattern is called optic flow (Gibson, 1950; Gibson, 1979). During translational movements a nearby object appears to move faster than its background. This apparent motion of objects during translational ego-motion provides spatial information about the environment, which can be exploited to construct a map based on the relative distances (Bertrand *et al.,* 2015; Faessler *et al.,* 2016) and to avoid collisions with obstacles while navigating through an environment (Kramer *et al.,* 1997; Zingg *et al.,* 2010; Clady *et al.,* 2014; Bertrand *et al.,* 2015; Milde, Bertrand, Benosman, *et al.,* 2015; Serres & Ruffier, 2017; Müller, Bertrand, *et al.,* 2018). Furthermore, optic flow has been used in conventional frame-based machine learning applications to segment images (Weinzaepfel *et al.,* 2013; Chen *et al.,* 2016), classify objects in videos (Rahtu *et al.,* 2010) or perform tracking (Manen *et al.,* 2014).

Most algorithms processing optic flow-based information rely on frames acquired from conventional imaging sensors. However, successive images in a video do not change at every pixel location. Thus, these algorithms perform unnecessary computation, due to the redundancy in the data. The computational cost can be lowered by using event-based vision sensors (Lichtsteiner, Posch, *et al.,* 2008; Posch, Matolin, Wohlgenannt, *et al.,* 2010; Brandli *et al.,* 2014; Posch, Serrano-Gotarredona, *et al.,* 2014; Son *et al.,* 2017). These sensors have the

advantage that only changes in temporal contrast, encoded as events, trigger the generation of data, thereby providing a sparse coding of the visual input (for more details see Section 1.1). Optic flow estimation can take advantage of this sparse representation, as already demonstrated by several studies (Benosman, Ieng, *et al.,* 2011; Benosman, Clercq, *et al.,* 2014; Conradt, 2015; Rueckauer & Delbruck, 2016; Liu & Delbruck, 2017), especially when the asynchronous encoding scheme is maintained, hence preserving precise timing, i.e. very low latencies, and '*pseudo-simultaneity*' (Camunas-Mesa *et al.,* 2012).

Unlike synchronous processing, which is performed in a serial manner (as in a CPU or microcontroller), asynchronous computing, similar to neural computing in the biological brain, naturally preserves temporal information without the need of explicitly encoding time, and offers the advantages provided by distributed and fully parallel computation.

Neuromorphic processing systems, especially mixed-signal (analogue/digital) ones, combine all the aforementioned properties and are thus well suited to operate on event-based data. Further advantages are provided by neuromorphic processors operating in the sub-threshold regime (Mead, 1989). The current flowing across a transistor which is operated in the sub-threshold range depends exponentially on the voltage supplied to the gate of the transistor. This exponential relationship is needed to model transfer characteristics found in biological neurons and synapses. In analogue sub-threshold neuromorphic processors, this exponential transfer characteristic can be implemented with a single transistor in contrast to digital super-threshold systems, in which a high computational load is required (Partzsch *et al.,* 2017). Neuromorphic sensory-processing systems are perfectly suited to be incorporated in autonomous agents in order to estimate optic flow from visual input. Such an agent could either be a flying, walking or wheeled robot, which should, depending on the field of application, be capable of navigating autonomously in any given environment. This kind of agent is required especially in the context of search and rescue missions, where not only size represents a crucial constraint, but also the payload (Calisi *et al.,* 2007; Ko & Lau, 2009). Especially the latter affects directly operation time of the agent. Asynchronous neuromorphic sensory-processing systems provide solutions that have been shown to scale with power consumption, in contrast to conventional synchronous GPU-based solutions (tens of mW vs. hundreds of W). Further, their inherent parallel computing architecture makes them even better suited for scaling up, due to their distributed computation.

The present work shows how a spike time-dependent gain modulation of the well-

known Differential Pair Integrator (DPI) synapse (Bartolozzi & Indiveri, 2007) adaptively scales the synaptic efficacy. This Adaptive Synaptic Efficacy (ASE) scaling enables a downstream neuron to encode the time-to-travel between two spatially adjacent inputs, e.g. pixels, into an instantaneous burst of spikes. The ASE circuit in combination with a DPI synapse and an adaptive, exponential leaky integrate-and-fire (LIF) neuron (Indiveri, Chicca, *et al.,* 2006) describe the spiking Elementary Motion Detector (sEMD) presented here. The time-to-travel is inversely proportional to the amplitude of optic flow. The size and duration of the burst produced by the LIF neuron directly reflects the temporal correlation of two spatially adjacent inputs, i.e. the closer in time the two spikes arrive relative to each other, the more the neuron spikes. We will show in detail how the sEMD can be modelled in software and how the principle can be further abstracted and emulated in mixed-signal neuromorphic circuits (hardware). Further, we will demonstrate that the sEMD can be used to extract a collision avoidance direction of a moving robotic agent in an outdoor cluttered environment. The proposed circuitry might constitute a possible connectivity scheme of how biological synaptic structures are organized, in order to estimate temporal correlation from discrete action potentials.

3.1.1 Event-based optic flow estimation

Event-based algorithms use temporal contrast changes to estimate optic flow. These changes are detected asynchronously by so-called event-based vision sensors, which send an event whenever the light intensity changes by a sufficient amount (see Section 1.1). Approaches to estimating event-based optic flow developed during the last three decades range from gradient-based method using the Lukas-Kanade (Lucas & Kanade, 1981) algorithm (Benosman, Ieng, *et al.,* 2011), local-plane fitting (Brosch *et al.,* 2015; Milde, Bertrand, Benosman, *et al.,* 2015), or relational networks (Martel *et al.,* 2015) to correlation-based methods based on either delay lines (Horiuchi, Lazzaro, *et al.,* 1991; Horiuchi, Bair, *et al.,* 1992), block matching of event-frames (Liu & Delbruck, 2017), or the time-to-travel algorithm (Kramer *et al.,* 1995).
This work proposes a novel correlation-based motion detection scheme for analogue Very Large Scale Integration (aVLSI) systems inspired by the time-to-travel algorithm (Kramer *et al.,* 1995). As stated earlier, the **time-to-travel** of events across the retina is inversely proportional to the relative velocity. Kramer and

colleagues (Kramer *et al.,* 1995, 1996, 1997) converted a fast brightness change into a single current pulse using a temporal edge detector circuit. In Kramer's temporal edge detector circuit, a current pulse was fed to a pulse-shaping circuit, which produced a slow monotonic decaying voltage signal. The respective pulse produced by a pixel x acts as a so-called *facilitation pulse*, while a pulse of the neighbouring pixel $x + 1$ triggers the measurement, i.e. the *trigger pulse*. The time-to-travel is directly encoded in the absolute output voltage of the circuit. The voltage is set by the relative time of the facilitation to the trigger pulse, and stored using a standard sample-and-hold circuit (Kramer *et al.,* 1997). As soon as the measurement is triggered the trigger pulse causes positive feedback, which ultimately resets the circuit to its resting state.

Conradt used the time-to-travel algorithm implemented on a **microcontroller** to extract optic flow directly from a **DVS** (Conradt, 2015). Events produced by the DVS are timestamped and used to compute the Δt between adjacent pixels in order to extract the time-to-travel. This approach has the advantage that motion estimation is possible within a wide dynamic range of velocities. However, it has the drawback that the time-to-travel is encoded as a fixed-point number, which cannot directly be used by a neuromorphic processor.

Giulioni and colleagues used an **event-based vision sensor** connected to a **mixed signal analogue/digital neuromorphic processor** in order to estimate optic flow in a more biologically plausible manner with low-power and low-latency requirements (Giulioni *et al.,* 2016). They used the same time-to-travel idea as originally proposed by (Kramer *et al.,* 1995). Their circuit motif can be summarized by feed-forward inhibitory connections to direction-selective neurons as identified by (Barlow & Levick, 1965). The time-to-travel is expressed by the number of spikes a so-called Elementary Motion Unit (EMU) emits. Four EMUs share one facilitation neuron which is connected with excitatory synapses to four direction selective cells (one for each cardinal direction). Each direction selective unit has one additional trigger neuron which is connected to an inhibitory synapse. The time for which one of the four direction-selective neurons is active with respect to the facilitation neuron's activity encodes the relative velocity of a stimulus.

We propose a novel motion extraction mechanism, which we call the spiking Elementary Motion Detector (sEMD). Similar to (Conradt, 2015; Giulioni *et al.,* 2016), we decouple the motion estimation from the sensor, in contrast to (Kramer *et al.,* 1995). This enables us to use various event-based vision sensors and identify the best suited one, given the constraints of our task. Conradt used

a synchronous processor, i.e. a microcontroller, to calculate the time-to-travel (Conradt, 2015). As argued earlier synchronous processors can in principle operate on event-based data, but the processor does not have the intrinsic distributed and parallel nature that would be needed to optimally exploit the sparsity and asynchronicity of event-based data. Like (Giulioni *et al.,* 2016), we emulated the motion detector in mixed-signal analogue/digital neuromorphic hardware and processed events coming from an event-based vision sensor using artificial neurons and synapses. By doing so we make the motion estimate easily available to a downstream network of spiking neurons. While (Giulioni *et al.,* 2016) needed 9 neurons and 8 synapses to realise a full motion detection unit, i.e. a 4-way motion detector, our implementation consists only of 4 neurons, 4 synapses, 12 additional transistors and 4 capacitors. Thus, the sEMD requires less silicon area for its implementation, while encoding the time-to-travel in the number of spikes of the neuron, similar to (Giulioni *et al.,* 2016). The aforementioned 12 transistors and 4 capacitors constitute 4 so-called Adaptive Synaptic Efficacy (ASE) circuits (see Section 3.2.2 for more details). This circuit adaptively scales the synaptic efficacy dependent on the relative timing of the trigger and the facilitation pulses. The essence of the computation of the proposed circuitry is to realize a temporal correlation detector, which is encoded by an instantaneous synaptic efficacy modulation. The synaptic efficacy modulation not only determines the absolute number of spikes produced by the neuron but also affects the interspike interval (ISI) distribution within a burst. In contrast to (Giulioni *et al.,* 2016), where the ISIs within a burst were kept constant, the ISIs within a burst in the sEMD response exponentially increase over time. This exponential increase enables the circuit to encode information about the time-to-travel already by the first two spikes of a burst. Thus, the sEMD can perform a fast but imprecise motion estimate with two spikes and provides a more accurate estimate within few milliseconds.

3.2 Methods

A DVS mounted on a mobile robotic platform emits events triggered by contrast changes produced by the boundaries of objects present in the environment (for details see Section 1.1). These events can be used to extract the relative motion of the objects due to the translational ego-motion of the camera (see Section 3.2.1).

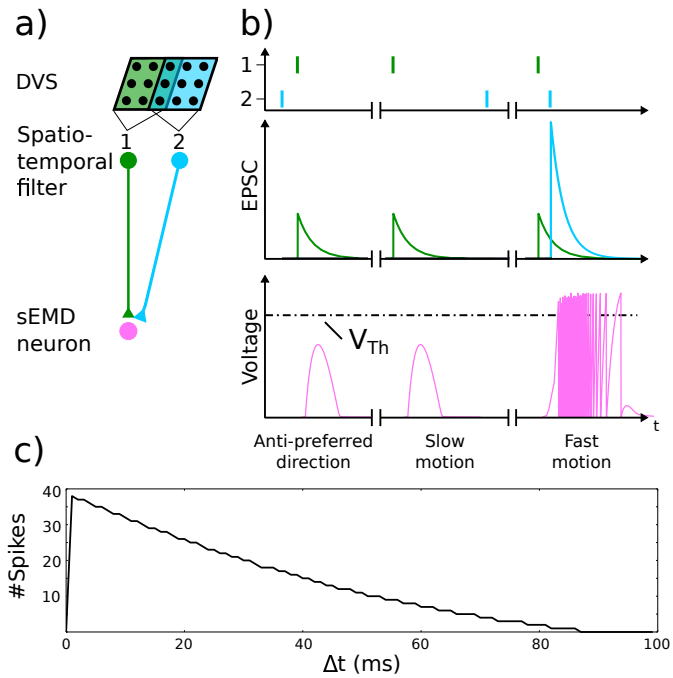


Figure 3.1: The spiking Elementary Motion Detector (sEMD) model in Brian2. a) 9 DVS pixels are connected to 1 spatio-temporal filtering neuron. The spatio-temporal filter neuron emits a single spike only if 6 or more DVS pixel are active within 35 *ms*. Only a single sEMD neuron is depicted. Both spatio-temporal filter neurons (facilitation, green) and (trigger, blue) are connected via a normal synapse to the sEMD neuron. b) The direction selectivity of the sEMD neuron is installed by modulating the synaptic efficacy of the trigger synapse (blue) by the Excitatory Post-Synaptic Current of the facilitation synapse (green) (see Equation 3.4). Only if neuron 1 spiked first and shortly before neuron 2, is the Excitatory Post-Synaptic Current produced by neuron 2 strong enough to elicit spikes in the sEMD neuron. c) Tuning curve of the sEMD. The number of spikes of the sEMD neuron distinguishes different Δt between facilitation and trigger spikes.

3.2.1 Spiking elementary motion detector in software

We designed a spiking neural network in Brian2 (Goodman & Brette, 2008) for implementing spiking elementary motion detectors compatible with event-based vision sensors. The network in its simplest form processes the sensor's events in order to (i) filter out noise and (ii) extract relative motion from the stream of events. Each neuron in the network is described by the differential equation of a simple linear leaky integrate-and-fire (LIF) neuron (Indiveri, Chicca, *et al.,*

2006) (Equation 3.1).

$$\frac{d\mathrm{V}}{dt} = \frac{1}{\mathrm{C}_m} \cdot (\mathrm{I}_e - (\mathrm{I}_{lk} + \mathrm{I}_i) \cdot (1 - e^{\frac{-\mathrm{V}}{\mathrm{U}_t}})), \tag{3.1}$$

where V is the membrane voltage of the neuron, t is time, C_m is the membrane capacitance, U_t is the thermal voltage, I_e is the excitatory current, I_i is the inhibitory current and I_{lk} is the leak current. Following each pre-synaptic spike the corresponding current is updated with a given synaptic weight w (Equation 3.2)

$$\mathrm{I}_{syn} = \mathrm{I}_{syn} + w. \tag{3.2}$$

All synaptic currents are described by a differential equation.

$$\frac{d\mathrm{I}_{syn}}{dt} = -\frac{\mathrm{I}_{syn}}{\tau}, \tag{3.3}$$

where I_{syn} is the synaptic current, t is time and τ is the time constant of the exponential decay of the Excitatory Post-Synaptic Current (EPSC).

Spatio-temporal filtering

The DVS response is noisy and, thus, needs to be filtered. This noise is, on the one hand, due to shot noise, which triggers events without a cause (Lichtsteiner, Posch, *et al.,* 2008; Yang *et al.,* 2015). On the other hand, the DVS is, like any neuromorphic hardware, prone to device mismatch[2] (Pelgrom *et al.,* 1989) and temperature sensitivity (Nozaki & Delbruck, 2017). The aforementioned sources of noise result in noise events and so-called hot pixels, i.e. pixels which continually emit events. To prevent the noise from altering the motion detection, the first layer of the network implements a spatio-temporal filter for removing events that are isolated in time and space. A spatial neighbourhood of 3 × 3 pixels of the DVS is selected. All pixels in one neighbourhood are connected to a single spatio-temporal filtering neuron, which generates a spike only if at least 6 local contrast changes within the neighbourhood are detected within 35 *ms* (Figure 3.1, a). The neuron's parameters are set to guarantee that, even when all pixels within the neighbourhood are active within 35 *ms*, only one output spike is produced (see Appendix, Table 7.4).
This layer not only cancels out noise but also reduces the amount of data to be

[2]Device mismatch is the response variability introduced to the electronic circuits due to imperfections in the manufacturing process. This leads to variability across all circuit blocks in an array.

processed by down-sampling the spatial resolution. The spatio-temporal filtering neurons help to ensure that the spikes fed into the sEMD neuron are triggered by a common cause, thus ensuring the presence of a change in contrast often due to a moving edge. The output of two neighbouring spatio-temporal filtering neurons is used to estimate the time-to-travel.

Measuring time-to-travel with spiking neurons

We propose a novel mechanism for coding time-to-travel with spiking neurons. To illustrate the principle, let us assume that we have two spatio-temporal filtering neurons and one sEMD neuron (Figure 3.1, a). The synaptic efficacy of the trigger synapse (blue) is determined by the EPSC of the facilitation synapse (green). The trigger synapse can only inject current into the neuron, i.e. produce an EPSC, if the synaptic current of the facilitation synapse is non-zero (Figure 3.1, b, compare anti-preferred direction vs. fast motion). Otherwise, the spike, which is propagated along the trigger synapse, has no effect on the post-synaptic membrane potential (Figure 3.1, b, slow motion and anti-preferred). Furthermore, the synaptic weight of the facilitation synapse is set to guarantee that, if there is no EPSC propagated along the trigger synapse, the sEMD neuron will not produce any output spikes (Figure 3.1, b, anti-preferred direction and slow motion). If the facilitating spatio-temporal filtering neuron spikes at t_0 the membrane potential rises. Further, if the trigger neuron emits a spike at $1\ ms < t_1 - t_0 < 80\ ms$, the EPSC of the facilitation synapse is non-zero (compare Figure 3.1 b, anti-preferred motion and fast motion). As a consequence, the sEMD neuron emits a burst of spikes, where the number of spikes encode the time-to-travel. The synaptic weight of the trigger synapse ($w_{e\,2}$) is multiplied by the synaptic current of the facilitation synapse ($\mathrm{I}_{e\,1}$) and added to the actual synaptic current of the trigger synapse ($\mathrm{I}_{e\,2}$):

$$\mathrm{I}_{e\,2} = \mathrm{I}_{e\,2} + (w_{e\,2} \cdot \mathrm{I}_{e\,1}) \tag{3.4}$$

This leads to a non-linear scaling of the EPSC (Equation 3.3) depending on the time-to-travel and in addition to providing motion perception it also gives us a direction selective mechanism for free. The sEMD neuron encodes the time-to-travel, which is represented by the Δt in the firing of the pre-synaptic spatio-temporal filtering neurons. This Δt is converted into an instantaneous burst of spikes (Figure 3.1, c). In our example a stimulus moving from right to left does not produce any output spikes, i.e. the membrane voltage stays

below the spiking threshold V_{Th} (Figure 3.1 b, anti-preferred). This is due to the multiplication of the synaptic weight $w_{e\,2}$ with the synaptic current $I_{e\,1}$, which is zero in this case.
We showed that the sEMD implemented in software is able to encode the time-to-travel of events produced by an edge moving in the visual field of an event-based sensor (Figure 3.1).

3.2.2 Spiking elementary motion detector in hardware

The sEMD features a direction-selective non-linear scaling mechanism of synaptic currents. The application of this motion-detection scheme to robotic sensory-motor tasks pose strong constraints on latency, power consumption and in some cases size/weight (e.g. in mini- and flying robots). We address these constraints by using sub-threshold neuromorphic solutions not only at the sensor but also at the computation level. The sub-threshold operation of the circuit produces very small currents (on the order of nano-amperes) resulting in low-power consumption. In this regime the transfer function of the transistor is exponential, therefore providing a powerful tool for the implementation of biological models at low "cost" (power-consumption and silicon area), given the otherwise high computational cost of the exponential function (Partzsch *et al.,* 2017). Therefore, we designed a circuit implementing the sEMD model described in Section 3.2.1 and fabricated it (Figure 3.2, left) using the standard Austria Micro Systems (AMS) 180 nm CMOS technology. The resulting chip comprises an array of 8 sEMD blocks (Figure 3.2, right). In order to stimulate and characterize the sEMD test chip we used the pyNCS framework (Stefanini *et al.,* 2014). The Differential Pair Integrator (DPI) silicon synapse (Bartolozzi & Indiveri, 2007), the short-term adaptation circuit proposed in (Ramachandran *et al.,* 2014) and the DPI adaptive, exponential LIF neuron (Indiveri, Chicca, *et al.,* 2006) are used as building blocks for the implementation of the sEMD model in hardware (Figure 3.2).

Adaptive synaptic efficacy circuit

Short-term plasticity, influencing the strength of the synapse (Markram, Pikus, *et al.,* 1998), can be classified by polarity: (i) short-term depression reduces the synaptic strength and (ii) short-term facilitation increases the synaptic strength. Circuits modelling short-term plasticity are normally connected to the gate of the weight transistor of a synapse. We, however, used a simple neuromorphic

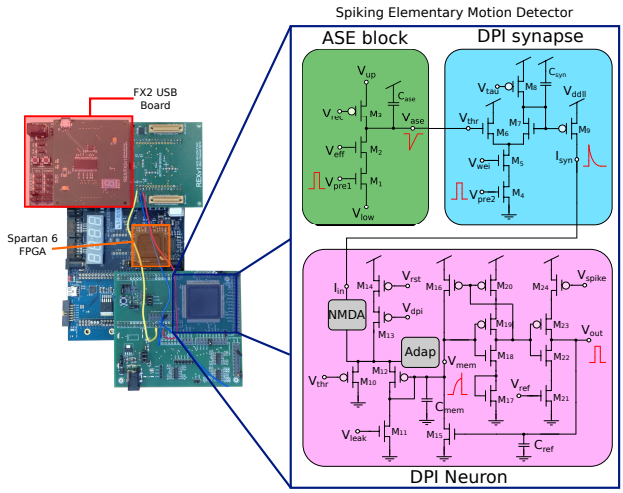


Figure 3.2: Left) Overview of the spiking Elementary Motion Detector (sEMD) test chip. The sEMD chip is interfaced by a Raggedstone 2 (*Raggedstone2 with Spartan 6 FPGA* 2017) board, containing a Spartan 6 Field Programmable Gate Array (FPGA). The FPGA is interfaced by the FX2 USB board (Fasnacht *et al.,* 2008) in order to set biases, stimulate and record activity. Right) Detailed schematics of the sEMD. The three computational blocks of the sEMD are shown in green (ASE), blue (DPI) and pink (neuron). The ASE block receives the 'facilitation' pulse (red) as input at the gate of M_1 and outputs the voltage V_{ase} to the gate of transistor M_6 (threshold) of the Differential Pair Integrator (DPI) synapse. The voltages V_{eff} and V_{rec} determine the amplitude and recovery rate of the V_{ase} voltage. The DPI synapse is turned on by receiving the 'trigger' pulse at the gate of M_4 transistor. The resulting output is the EPSC I_{syn} whose amplitude is the set by the voltages V_{wei} and V_{ase}, and V_{tau} controls the time-constant of recovery. The output current I_{syn} is injected into the neuron (I_{in}). The neuron integrates the incoming current and starts spiking once the membrane potential rises above its threshold set by the voltage V_{thr}. The leak and the refractory period of the neuron can be set by the corresponding voltages V_{leak} and V_{ref}. The NMDA-block and the Adaption-block are turned off and not used in this study. For details see (Indiveri, Chicca, *et al.,* 2006).

short-term depression circuit presented in (Ramachandran *et al.,* 2014) to alter the synaptic efficacy independent of the weight. Hence we will call this block Adaptive Synaptic Efficacy (ASE) circuit hereafter. In response to the incoming spike (event/pulse), the ASE circuit alters the synaptic efficacy and offers an independent control over the recovery rate of the efficacy. In our architecture, the ASE circuit receives the first input event and thus facilitates the motion estimate. The ASE circuit features a weight, w_{ase}, as well as a time constant, τ_{ase} set by V_{eff} and V_{rec} respectively (Figure 3.2). Both biases affect the output

voltage of the ASE circuit in response to a pre-synaptic spike. The output voltage of this circuit is provided as a gate voltage to the threshold transistor of the DPI synapse as shown in the block diagram (Figure 3.2).

Synapse circuit

The DPI synapse presented in (Bartolozzi & Indiveri, 2007) is one of the most used synapses in sub-threshold neuromorphic chips. In response to the pre-synaptic input spikes (events/pulses), the DPI synapse gives an exponential decaying EPSC as its output depending on the parameter setting (Figure 3.2). The weight, w_{dpi} which is set by V_{wei}, of the synapse, determines the amplitude of the EPSC evoked during a pre-synaptic spike. Whereas the time-constant τ_{dpi}, which is set by V_{tau}, dictates the temporal evolution of the EPSC in between the pre-synaptic spikes. The DPI synapse has a threshold bias. The transistor which sets the synapse's threshold is usually employed to implement any global computation that affects the EPSC of the synapse, such as homoeostasis mechanism. But in fact, a voltage supplied to the gate of the threshold transistor scales the amplitude of the resulting EPSC along with the w_{dpi} associated to the synapse circuit (Figure 3.2). To realise a quarter-way motion detector we used one DPI synapse which acts as a trigger pulse generator. To facilitate a motion estimate we use an ASE block which is connected to the threshold of the corresponding synapse.

Neuron circuit

We used the adaptive exponential LIF neuron circuit presented in (Indiveri, Chicca, *et al.,* 2006). The neuron integrates the incoming synaptic current which charges its membrane capacitor C_{mem} (Figure 3.2). Furthermore, the silicon neuron offers a tunable leakage current, as well as a spiking threshold. All biases to tune the neuron's behaviour can be set externally and are stored on chip, once they are loaded. If the membrane voltage surpasses the spiking threshold value, a positive feedback loop is activated, during which C_{mem} is quickly charged. Consequently, the neuron consumes very little power during the spike generation (Indiveri, Chicca, *et al.,* 2006). Right after the spike generation, the membrane potential is reset to zero by enabling the reset transistor (M_{15}) and then the refractory period kicks in. The length of the refractory period is also determined by an external bias (V_{ref}) which sets the gate voltage of the refractory transistor M_{21}. The current through the refractory transistor discharges C_{ref} and slowly

turns off the reset transistor M_{15}, thus preventing the neuron from eliciting another spike. In our implementation, the refractory period is set to ∼1 ms (see Appendix Table 7.5 for detailed list of parameters used in this study). We tuned the parameters of the neuron so that no spike is elicited in response to small EPSCs, however, the neuron spikes for large input currents.
The ASE circuit determines the amplitude of the EPSC generated by the DPI synapse, that causes spiking output of the neuron. Therefore, the firing-rate of the neuron is determined by the threshold value of the synapse, which in turn depends on the time-to-travel of the event across the focal plane. It should be noted that the DPI synapse receives events independent of the ASE circuit. In this way, the sEMD circuit obtains a direction-selective response and encodes the time-to-travel in its spiking output.

3.3 Results

To characterize the circuit's response to simple and well controlled stimuli we used a single facilitation pulse and (in contrast to typical usage) multiple trigger pulses (as defined in Section 3.1.1) with different timing as shown in Figure 3.3. Before the facilitation pulse is provided, the output voltage of the ASE circuit (V_{ase}) is at its resting level (0.9 V) (Figure 3.3, ASE). As soon as the facilitation pulse is sent to the gate of M_1, the ASE circuit's capacitor (C_{ase}) is quickly charged and the ASE circuit's output voltage drops to 0.4 V (Figure 3.3, ASE). The size of the voltage drop is set by the gate voltage of the M_2 transistor. As soon as V_{out} deviates from its resting level, the current through the M_3 transistor starts discharging the capacitor again. The time constant of the recovery of V_{out} to its resting voltage is set by the gate voltage V_{rec} of M_3. The output voltage of the ASE circuit is connected to the gate of the threshold transistor (M_6) of the DPI synapse (Figure 3.2), therefore modulating the synaptic efficacy. The trigger pulses are provided to the gate of M_4 of the DPI synapse at 8 different times relative to the facilitation pulse (Figure 3.3, AER Input, Δt = 2, 12, 22, 32, 42, 52, 62, 72 *ms*). The relative time between the facilitation and the trigger pulse represents the time-to-travel (for more details see Section 3.1.1). The trigger pulse is used as input to the DPI synapse, therefore generating an output current which is a function of the output voltage (V_{ase}). A short time-to-travel results in a low V_{ase} at the time of the trigger pulse, which in turn produces a large change in the DPI's output voltage. Hence, the amplitude of the output EPSC is also large. A longer time-to-travel produces a smaller EPSC amplitude.

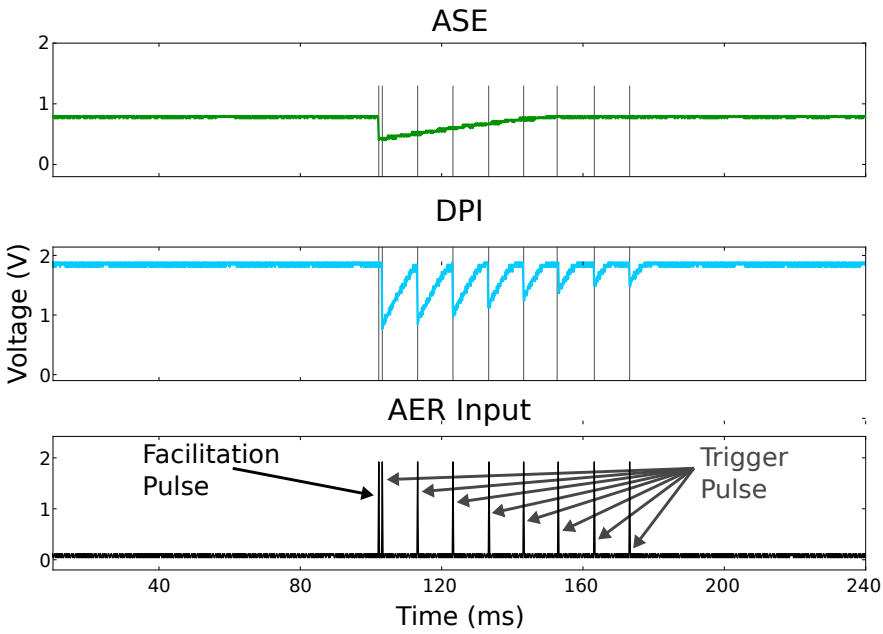


Figure 3.3: Adaptive Synaptic Efficacy (ASE) circuit's response and its effect on the DPI's efficacy. Voltage traces of both circuit blocks in response to different Δt. The synaptic weight, w_{dpi} set by the gate voltage of M_5, is unaltered. The synaptic efficacy is determined by the threshold voltage which in turn is set by the ASE's output voltage. The biases of the sEMD circuit were set to emphasise its response. The DPI time constant is usually in the range of few tens of milliseconds.

The increase in amplitude of the resulting EPSC follows a multiplicative effect on top of the amplitude set by gate voltage of M_5 (V_{wei}) of the synapse. The time-constant of EPSC is determined by the gate voltage of M_8 (V_{tau}) of the synapse and is unaffected by the V_{ase}. Therefore, the downstream neuron can integrate more current and thus elicit more spikes.

To show this effect we stimulated the circuits with two time-to-travel (Δt = 2, 20 *ms*) and observed the neuron's spiking behaviour in terms of number of spikes elicited, as well as the duration of the burst (Figure 3.4, Neuron). The gate voltage of M_6 (Figure 3.4, ASE) is at 0.45 V and 0.65 V when the respective trigger pulses arrive at the DPI synapse. Thus, the resulting EPSC amplitude is larger at Δt = 2 *ms* compared to Δt = 20 *ms* (Figure 3.4, DPI). Since the ASE circuit is not connected to the neuron, the facilitation pulse has no effect on the membrane potential (Figure 3.4, Neuron). Only the DPI can inject current into the neuron when M_4 receives the trigger pulse at its gate. The larger EPSC

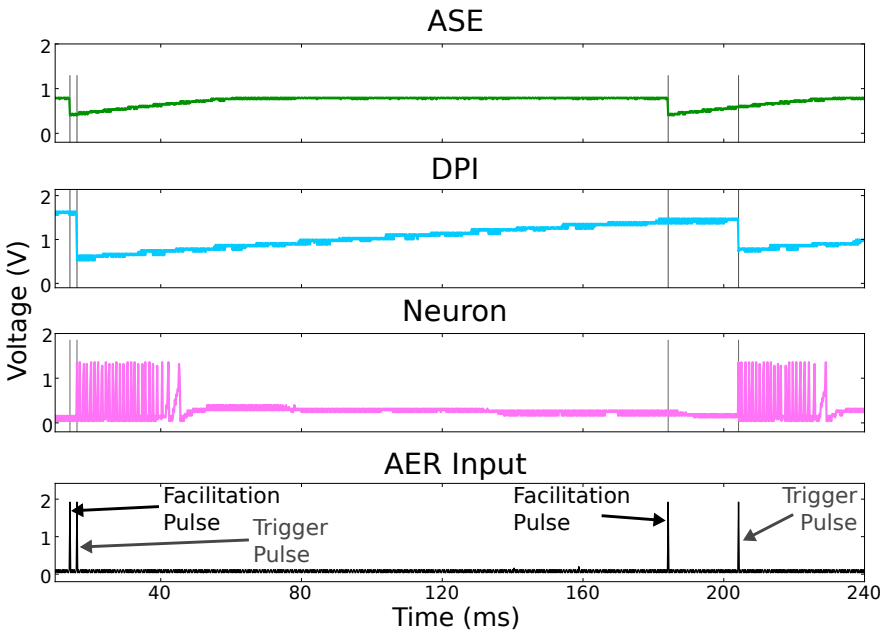


Figure 3.4: Membrane potential of an example neuron in response to two different Δt between facilitation and trigger pulse. Note that not only the number of spikes but also the burst length is set by the relative timing of the pulses. The biases of the sEMD circuit were set to emphasise its response. Burst duration usually does not exceed 10 *ms* and the DPI time constant is in range of few tens of milliseconds.

amplitude for smaller Δt results in more spikes (26 within 35 ms at $\Delta t = 2$ compared to 22 within 25 ms at $\Delta t = 20$).

We conducted this experiment to demonstrate the mode of operation of the sEMD in its simplest form and to provide an intuitive understanding of the impact of different voltage levels on the sEMD's response. The amplitude of the EPSC which is provided to the neuron as I_{in} was found to be scaled depending on the output voltage V_{ase}, as intended by design. The gate voltage V_{tau} of M_8 is fixed and thus the amount of current which is injected into the neuron is only determined by the amplitude of the EPSC. The neuron integrates this current and varies its response in terms of the number of spikes accordingly.

To further characterize the sEMD circuit and obtain the full tuning curve (neuron response vs. time-to-travel) we systematically varied the relative time between the facilitation and trigger pulse (Δt) and measured the number of spikes within a burst, the burst's duration, as well as the interspike interval (ISI) distribution within a burst.

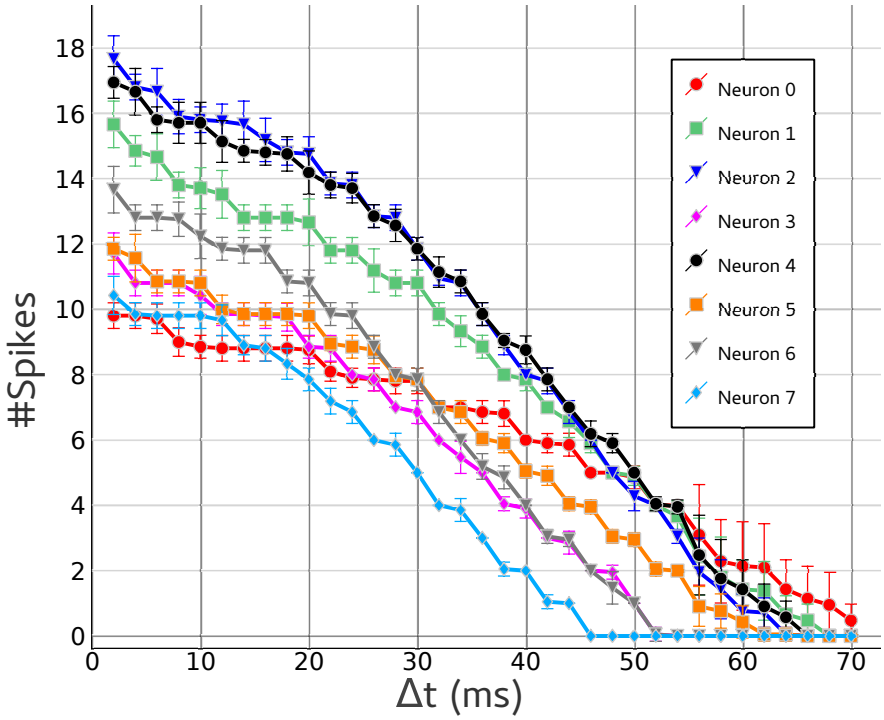


Figure 3.5: Tuning curves of the sEMDs. Number of spikes within a burst for each sEMD neuron is calculated for 20 stimulus presentations. The respective mean and standard deviation is calculated. The mean represents the deviations in responses across the neuron array resulting from device mismatch in our chip. The error bars indicate intra-neuron response variability and could be due to thermal noise. The intra-neuron response variability is less than 0.4 spikes during a burst.

The relative timing of the two pulses was varied from 2 *ms* to 70 *ms* with a 2 *ms* step size. The biases, which are shared among all 8 sEMD circuits, were tuned for neuron 0 and kept fixed through all recordings (a detailed list of biases used can be found in the Appendix, Table 7.5). Most sEMD neurons show a flattened tuning profile for short Δt, whereas all of them exhibit a linear encoding for intermediate Δt and a saturation region for large Δt (see Figure 3.5). All 8 test circuits consistently show higher spiking activities for smaller Δt, in contrast to larger Δt which generate fewer spikes. We find that the slope for small time-to-travel (approximately smaller than 20 *ms*) is less steep compared to the slope for intermediate time-to-travel. This provides an optimal resolution at the operation regime of the circuit while maintaining a wide dynamic range. It should be noted that the operation range is mainly set by the time constants of

the ASE circuit and the DPI synapse, V_{rec} and V_{tau} respectively, which makes it easy to adjust the dynamic range to meet the operation range needed for any given task. With the current bias setting the sEMD has a dynamic range of 34 dB, or in other words, 2.4 - 85 $\circ/s$. Overall we tested the circuit with a variety of biases and we were able to distinguish Δt ranging from 10 *ns* up to hundreds of *ms*. The time constants of the ASE and DPI circuit ultimately determine the current which is injected into the post-synaptic neuron. The refractory period of the neuron defines the maximum number of spikes and the shortest interspike interval the neuron could potentially generate given the input current. Thus, the refractory period and other parameters, such as the neuron's leakage or threshold, modulate the sEMD's response. The instantaneous frequency, which is defined as the number of spikes within a burst divided by the duration of the burst, is not informative since the neuron is only sparsely active and only for a very short amount of time (< 10 *ms*).

To characterize the variability of the responses due to thermal noise, we calculated the mean and standard deviation of each neuron across 20 stimulus sweeps (Figure 3.5). We also looked into the variations in the circuit responses across the array due to device mismatch effects and plotted the population tuning curve (Figure 3.6). It is worth mentioning that the circuit blocks 0 and 7 are physically located at the border of the silicon area and the mismatch tends to be larger in these areas due to corner effects (Pavasović *et al.,* 1994). The effect of mismatch is clearly visible in neuron 0 and 7, which show different profiles of spiking activity to Δt compared to the rest (Figure 3.5).

The instantaneous bursting response shows clearly how the precise timing of incoming spikes can be translated in order to further process the motion information. We operate the circuit and its transistors in the sub-threshold regime which, as stated earlier, yields an exponential relationship between the gate voltages and the current flowing through each transistor. Additionally, the efficacy modulation of the synapse follows a multiplicative effect. We expected to see this non-linear scaling reflected in the tuning profile of the sEMD, however, we could only find it slightly reflected at the population response level. We thus investigated the burst more carefully in terms of its ISI distribution and found that the non-linear scaling is indeed preserved in the temporal evolution of the ISI within a burst (Figure 3.7).

A very short time-to-travel (Δt < 20 *ms*) saturates the neural response to its refractory period. After a few spikes into the burst, the ISI tends to increase exponentially until all provided current is integrated and translated into spikes.

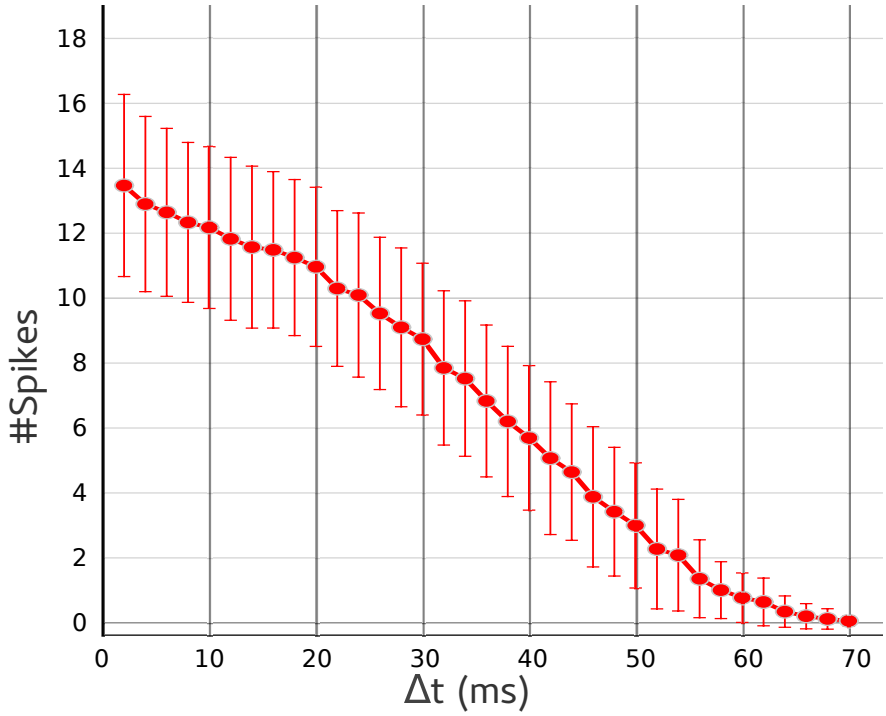


Figure 3.6: Tuning curve of the sEMD population. Population response of an array of 8 sEMD neurons averaged over 20 stimuli presentations and 8 neurons. The mismatch, i.e. inter-neuron variability, represented by the error bars (∼2 spikes), is larger than the intra-neuron response variability (0.4 spikes). The population tuning curve, in contrast to a single tuning profile, shows an exponential decrease for large Δt which results from mismatch. For intermediate Δt the tuning curve becomes linear and steeper compared to short Δt. The different slopes of the tuning curve offer good velocity resolution, while preserving high dynamic range.

The ISI increases exponentially over time, longer time-to-travels elicit larger ISIs, fewer spikes, and shorter burst durations. For longer time-to-travel ($\Delta t > 20$ *ms*) the first two spikes carry enough information to estimate motion on a coarse scale (Figure 3.7, grey and black curve). This way of encoding information in the neuronal response has not been exploited so far by any studies to our knowledge. We hypothesize that the two described information encoding schemes, namely the number of spikes within a burst and the ISI distribution, can be seen as complementary, and potential benefits from this superimposed scheme are discussed below (see Section 3.5).

To summarize we have shown that the sEMD mechanism can be implemented

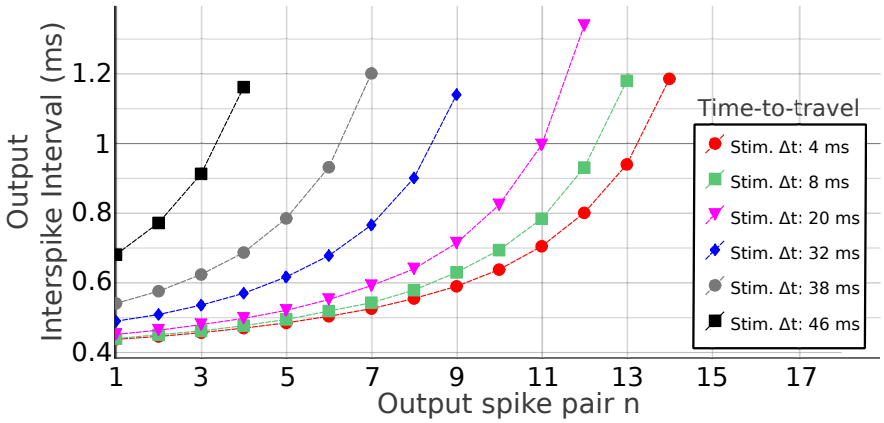


Figure 3.7: Interspike interval (ISI) distribution within a burst. ISI distribution for six different time-to-travel (4 *ms*, red dot; 8 *ms*, green square; 20 *ms*, pink triangle; 32 *ms*, blue diamond; 38 *ms*, grey dot; 46 *ms*, black square) within the burst of activity. The ISI is calculated as $\mathrm{ISI}_n = t_n - t_{n-1}$. Thus, the x-axis indicates the respective spike pair. The first three stimuli (red dots, green squares, pink diamonds) make the underlying neuron fire at its maximum frequency right after the stimulus onset. However, the ISI tends to increase faster for longer Δt's compared to smaller ones. Note that the neuron's activity triggered by longer Δt (blue diamonds, grey dots, and black squares) is slower (longer ISI of the first two spikes). This suggests that the precise timing is not only encoded in the instantaneous spiking activity, but also in the relative distribution of the spikes within the burst and that the first two spikes are already informative about the presented velocity.

in a mixed-signal analogue/digital neuromorphic chip, and we characterised it both in software and in hardware implementations in terms of single neuron responses. Our sEMD responds to any two events occurring spatially and temporally close to each other, even if the two events are not linked to physical motion. Second-order motion stimuli give, for example, an impression of movement although nothing is physically moving. This class of stimuli results from changes in contrast, texture or some other quality that does not result in an increase in luminance or motion energy in the Fourier spectrum of the stimulus. Animals usually respond to second-order stimuli (e.g. fly (Theobald *et al.,* 2008), monkey (O'Keefe & Movshon, 1998), human (Nishida *et al.,* 2003)). A large number of second-order motion stimuli exist, but are not directly relevant for a real-world tasks, such as collision avoidance, since for such a task objects physically move. Thus, the systematic investigation of the response of the sEMD to second-order motion goes beyond the scope of this paper. Nevertheless, we can anticipate the

response of the detector for a simple type of second-order motion, the Mu-line. For a Mu-motion (Lelkens & Koenderink, 1984), for each frame, a successive column of pixels is refreshed by random values (dark or bright). Our sEMD will interpret this as a time-to-travel if the pixel brightness of a given column and the successive column changes, and not otherwise. In the next section, we test this computational block in a real-world task, using a computer simulation of the sEMD, and testing the motion estimation mechanism on a moving wheeled robotic platform in open-loop.

3.4 Collision avoidance in outdoor cluttered environments

Since the current implementation on the test chip of the sEMD circuit only had 8 circuits, we used computer simulations (see Section 3.2.1) to verify the working behaviour of our motion detection model in a real-world open-loop robotic scenario.

A robot was remote controlled and steered on a straight line in the centre between obstacles placed in an outside environment (see Appenix, Figure 7.1). A standard webcam (Figure 3.8, first column) and a DVS (Figure 3.8, second column) was mounted on the robot recording data during translational movement (Figure 3.8, row 1 - 3). Positive and negative contrast changes, i.e. ON- and OFF-events from the DVS, were elicited by boundaries of objects. The data file sizes of the events and the webcam images were 1 MB and 9 MB, respectively, with a total recording time of 10 seconds. [3]. The difference in file size illustrate the advantage of event vs. frame-based methods in terms of the amount of data being generated and processed. To emphasise the edges of obstacles and reduce noise events as well as texture-induced events, we connected the DVS output to a layer of spatiotemporal filtering neurons (Figure 3.8, third column and see Section 3.2.1 for details). The activity of this layer was used by the sEMD neurons to extract optic flow (Figure 3.8, fourth column) along the horizontal axis. The sEMD is capable of estimating rotational optic flow, due to e.g. rotational ego-motion. However, rotational optic flow only scales with the angular velocity and does not provide information about the distance to objects (Egelhaaf *et al.,* 2014). This information is crucial in the context of collision avoidance tasks; therefore, only translational optic flow can be used for ensuring collision-free

[3]The webcam recorded 25 frames per second and each frame was stored as PNG. The events are stored as a 32 bit floating-point python numpy array. No compression was used to store the events (.npy file format).

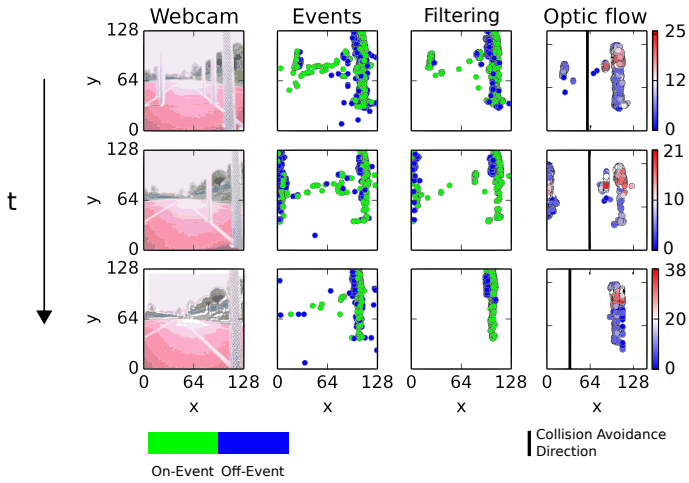


Figure 3.8: Neuron activity of the collision avoidance network in an open-loop for three different time points. The robot moves in an outdoor environment; here eight objects were present. The time increases from top to bottom row. First column: Webcam frames. Second column: The projection of the events within a time $t_i \pm 35$ ms on the $(x-y)$ plane, i.e. the camera coordinates. ON- and OFF-events are represented by green and blue dots, respectively. Object boundaries can be seen by neighbouring on and off edges. Third column: Spatio-temporal filtering of the events detected by the DVS. Fourth column: Spiking activity of sEMD neurons and extracted collision avoidance direction. Number of spikes of activated sEMD neuron is colour coded. Output of sEMD layer is vertically integrated, e.g. along the y-axis. The averaged activity is used to suppress the inverse soft winner-takes-all. The soft inverse winner represents the collision avoidance direction (black bar). Note that the collision avoidance direction points always away from close by objects along different points of the trajectory.

movements. Each sEMD neuron encoded the time-to-travel across the retina into an instantaneous burst of spikes. The sEMD responded to time-to-travel from 1 *ms* to 80 *ms*. The lower limit was due to the simulation time step of 1 *ms*. The upper limit is set by the chosen parameter setting (see Appendix Table 7.2.4). The maximal number of spikes within a burst was 38. The sEMD neurons emitted more spikes when the time-to-travel was short.

To test sEMD responses in the context of robotic navigation we projected the neuronal activity to a WTA. Since a high input rate to the WTA signals a close-by object, given constant translatory ego-motion, we reversed the WTA response to select the inputs with the lowest activity. We call this a soft inverse WTA layer, similar to (Horiuchi, 2009), in order to extract a collision avoidance direction. The relative position of an object in the visual field, as well as its relative near-

ness, was used to suppress neuronal activity in the soft inverse WTA. The activity in the soft inverse WTA layer is used to determine a steering direction, i.e. a collision avoidance direction (Figure 3.8, fourth column black bar). To evaluate the performance of the sEMD more systematically, we compared the output of the soft inverse WTA using the sEMD (Figure 3.9, dot-dashed orange line) as described above with the Center Of Mass Average Nearness Vector (COMANV) algorithm (Bertrand *et al.,* 2015) along the entire robot's trajectory. The COMANV algorithm has been shown to successfully estimate the collision avoidance direction in open- as well as closed-loop scenarios using walking and wheeled robots (Bertrand *et al.,* 2015; Milde, Bertrand, Benosman, *et al.,* 2015; Meyer *et al.,* 2016). We used two different inputs to the COMANV algorithm: (i) the output of an array of conventional EMDs (Meyer *et al.,* 2016) which extracted optic flow from the webcam images (Figure 3.9, solid dark-grey line) and (ii) the output of the sEMD array (Figure 3.9, dashed blue line).

To asses the performance of each of the aforementioned approaches and make comparison easier, we calculated the collision avoidance direction in image coordinates. Since the robot was remotely controlled and steered in the centre of the obstacle corridor with a slight left bias due to the objects' arrangement (see Appenidix, Figure 7.1), a reasonable collision avoidance direction would point right along the middle of the obstacle corridor, i.e. $\approx$ 64th pixel. We rescaled the collision avoidance direction (CAD) to be 0, meaning that the robot would move straight ahead, i.e. $\text{CAD}_{rescale} = 64 - \text{CAD}$. A positive collision avoidance direction indicates that the robot would steer to the left, whereas a negative collision avoidance direction indicates that the robot would steer to the right if it were to be operated in closed-loop.

After the robot starts moving, the sEMD + WTA shows huge fluctuations in its collision avoidance direction, which are due to the background input activity of the WTA population consisting of a Poisson spike train. As soon as the robot starts entering the obstacle corridor, indicated by the grey box, the estimated collision avoidance direction converges towards zero, with a slight bias towards positive collision avoidance directions. After leaving the obstacle corridor the noise in the Poisson input dominates again the collision avoidance direction, due to missing obstacles in the visual field. The fully conventional approach (solid dark-grey line) does not suffer from noisy input, nor noisy computation, but shows rather a smooth collision avoidance direction especially when moving along the obstacle corridor. The maximum difference in collision avoidance direction between the fully conventional and the fully asynchronous approach

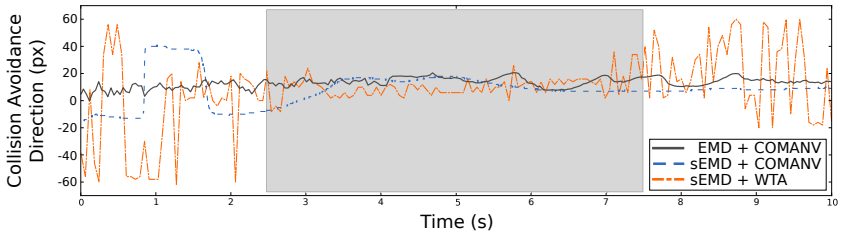


Figure 3.9: Comparison of conventional frame-based (solid dark-grey), hybrid (dashed blue) and fully asynchronous event-based (dot-dashed orange) collision avoidance systems. Due to the retinotopic organization of the motion detector arrays, the collision avoidance direction is given in pixels along the horizontal axis of the camera. A collision avoidance direction of zero indicates a straight forward motion. The obstacle corridor is indicated by the grey box. The conventional approach (solid dark-grey) uses the well-established motion detection model EMD as proposed by (Hassentstein & Reichardt, 1956) and implemented by (Meyer *et al.,* 2016) and the Center Of Mass Average Nearness Vector (COMANV) algorithm (Bertrand *et al.,* 2015). The hybrid approaches uses optic flow as extracted by the sEMD and the COMANV algorithm to determine the collision avoidance direction. All three collision avoidance approaches estimate a reasonable collision avoidance direction as soon as the robot enters the obstacle corridor. The huge fluctuations in collision avoidance direction of the fully asynchronous event-based approach (sEMD + WTA, dot-dashed orange line) before and after the obstacle corridor are due to the Poisson noise which biases the WTA and indicates missing prominent visual input, i.e. objects are too far away.

is less than 10 pixels. The small bias towards positive collision avoidance directions, i.e. to the left, at the end of the obstacle corridor is due to the different size in the field of view of the DVS and the webcam. While the last object on the right was already at the right border of the visual field of the webcam, the object's position in the field of view of the DVS was rather more centric (compare Figure 3.8, last row first and second column). The combination of asynchronous motion perception using the sEMD and the COMANV algorithm shows the best collision avoidance direction, apart from a brief period before entering the obstacle corridor. This big deviation might be due to noise in the visual sensory input stream (see. Section 3.2) or to the activity onset of sEMD neurons. Overall, all three approaches are able to predict a collision avoidance direction which ensures a collision-free navigation path within the obstacle corridor. The collision avoidance direction as extracted by the spiking neural network performs equally well as conventional frame-based approaches, but it is suitable for a neuromorphic implementation based on sub-threshold analogue circuits. Such implementation

offers several advantages including significant lower power consumption and much higher temporal resolution, while producing a sparse coding of the output. Thus, the fully asynchronous event-based approach might be worth pursuing in the context of closed-loop collision avoidance in cluttered environments. Closed-loop control systems, that rely on neuromorphic sensory systems, have been proposed in the last decades ranging from vision-based pencil balancing (Conradt, Cook, *et al.,* 2009) to auditory-based source following (Klein *et al.,* 2015). However, post-processing of sensory information was done with conventional CPUs and the sensor itself was stationary. Whilst tremendous research effort has been put to post-process the sensory information on neurally-inspired hardware, such as neuromorphic processors, to model saccadic eye movements and visual attention (Horiuchi, Bishofberger, *et al.,* 1994; Indiveri, 2001; Bartolozzi & Indiveri, 2009), the sensor, which was mounted on a stationary pan-tilt unit, was not translating in the environment. In the domain of closed-loop neuromorphic robotic navigation application researchers have either used conventional cameras and post-process the sensory information by neuromorphic processors (Hwu, Isbell, *et al.,* 2017; Hwu, Krichmar, *et al.,* 2017) or used neuromorphic vision sensors, but post-processed the sensory information using conventional CPUs (Indiveri & Verschure, 1997; Hoffmann *et al.,* 2013; Clady *et al.,* 2014; Luber *et al.,* 2015; Moeys *et al.,* 2016). So far, fully neuromorphic, i.e. neuromorphic sensory-processing, closed-loop robotic navigation systems have only been shown recently to work in real-world scenarios (Denk *et al.,* 2013; Galluppi *et al.,* 2014; Milde, Blum, *et al.,* 2017). However, these systems rely on hand-engineered features to navigate, e.g. blinking LED or laser-pointer, and do not use information about the object's distance while navigating. Whilst closed-loop implementation is beyond the scope of this paper, however, we already implemented the sEMD on a SpiNNaker (Furber, Galluppi, *et al.,* 2014) board and we plan to set up a closed-loop experiment in the future (see Appendix, Figure 7.3).

3.5 Discussion

We presented the spiking Elementary Motion Detector (sEMD) and its application to the extraction of motion information from event-based vision sensor data. We showed how the mechanism can be simulated with spiking neurons in software, emulated in mixed-signal neuromorphic hardware and implemented on a digital neuromorphic processor (see Appendix, Figure 7.3). We characterised the silicon sEMD neurons and showed that their output reliably encodes time-to-travel. The

software implementation revealed that the proposed mechanism can be used to extract a collision avoidance direction in the context of robotic navigation.
The sEMD encodes the time-to-travel of an event travelling across the retina in (i) the absolute number of spikes within a burst and (ii) the ISI distribution within a burst. The novelty of the sEMD is two-fold. First, we use the sparse bursting behaviour of a neuron to provide a fast response lasting less than 10 milliseconds. Second, the sEMD encodes information about the velocity already in the first two spikes of the burst (Thorpe, Fize, *et al.,* 1996; Thorpe, Delorme, *et al.,* 2001; VanRullen *et al.,* 2005). These two spikes can be used by a network to distinguish fast vs. slow motion on a coarse scale in a quick way. In the next 10 *ms* an accurate velocity estimation can be obtained from the ISI, because the absolute number of spikes within a burst depends ultimately on the time-to-travel (Figure 3.5 & 3.7). This way of encoding the velocity estimation is possible because we modulate the synaptic efficacy based on the precise spike timing. The silicon implementation of the sEMD presented in this work consist of a first prototype comprising 8 cells, designed for testing the functionality of the circuit. Given the very good results obtained, we are now in the position to design a large-scale array of sEMD circuits for building multi-chip systems suitable for autonomous navigation. The design will require a thorough analysis of device mismatch through Monte-Carlo simulations to allow the implementation of consistent motion flow estimation within the array.

State-of-the-art in event-based optic flow estimation

The proposed sEMD mechanism is a correlation-based motion detection scheme which relies on the precise timing of neighbouring pixels. It measures the time an event needs to travel across the retina, i.e. the time-to-travel. Our goal was (i) to efficiently estimate the time-to-travel from event-based vision sensors and provide a scalable and highly flexible solution by decoupling the estimation from the sensor. In order to optimally exploit the sparsity and asynchronicity of event-based data, we wanted (ii) an asynchronous solution which features additionally a distributed and parallel processing scheme with low power consumption and low latencies. To enable a downstream spiking neural network to further process the motion information we needed (iii) an implementation on a mixed-signal neuromorphic processor by estimating the motion using artificial spiking neurons and synapses. Last but not least, our objective was to (iv) build a circuit which can operate on a very fast timescale in order to be used in the

context of robotic navigation.

The time-to-travel algorithm was originally proposed by (Kramer *et al.,* 1995). Kramer's facilitate-and-sample circuit encoded the time-to-travel using a constant voltage level (Kramer *et al.,* 1995, 1997). Furthermore, the temporal edge detector was co-located on the same chip. The constant voltage level does not directly enable a downstream network of spiking neurons to further process this information and the co-location of pixel and motion detector prevents the system from being scalable without a new chip.

Conradt's (Conradt, 2015) implementation of the time-to-travel algorithm on a microcontroller overcomes the scalability issue by using a separate vision sensor but avoids a neural implementation by calculating the differences between neighbouring event timestamps. This implementation also does not address the problem of how to further process the motion estimate in a network of spiking neurons.

Giulioni and colleagues (Giulioni *et al.,* 2016) proposed for the first time an event-driven motion estimation approach, with decoupled sensor and processor, which allows scalability and encoding of time-to-travel in neuronal activity suitable for a downstream spiking neural network. One so-called Elementary Motion Unit (EMU) is implemented with 2 neurons and 2 synapses, plus 1 facilitation neuron which is shared among 4 EMUs. The time-to-travel is only encoded by the absolute number of spikes and not in the first ISI, thus their system needs to wait till the respective stop neuron inhibits the direction-selective neuron. In contrast to Giulioni's work a single sEMD only needs 1 neuron, 1 synapse, and 1 ASE circuit (3 transistors and 1 capacitor). The sEMD encodes the time-to-travel partially by the number of spikes within a burst of activity, but in addition, the ISIs within a burst of the sEMD increase exponentially and the first ISI already provides a coarse motion estimate.

In conclusion, the present work proposes a novel event-based motion detection scheme, called the spiking Elementary Motion Detector. The sEMD circuit requires less silicon area to be implemented compared to other works and encodes the time-to-travel by the absolute number of spikes (providing precision) as well as by the ISI distribution (providing low latency). When a decision has to be made quickly, relative timing of the first two spikes can be used to generate a fast, coarse motion estimate. The motion can then be more accurately estimated over the next 10 *ms* from the absolute number spikes within the burst.

State-of-the-art collision avoidance

Our proposed spiking neural network with the sEMD model implemented to estimate optic flow from an event-based camera and an inverse soft winner-takes-all (WTA) population to estimate a collision avoidance direction works as well as conventional frame-based approaches (Meyer *et al.,* 2016) in an open-loop setting. These results suggest that the proposed approach might also be useful in a closed-loop scenario as it has been shown already for the frame-based approach (Meyer *et al.,* 2016). However, closed-loop analogue neuromorphic sensory-processing systems, in which the spiking activity directly affects the robots steering behaviour, have only recently been shown to be able to navigate in real-world scenarios (Milde, Blum, *et al.,* 2017; Milde, Dietmüller, *et al.,* 2017). A digital neuromorphic processor, i.e. SpiNNaker, has already been shown to successfully steer a robot and perform target acquisition and following (Denk *et al.,* 2013). The aforementioned studies only used simple hand-engineered features such as tracking a certain LED's flickering frequency or counted the number of spikes in certain sub-regions of the visual field to generate histograms, that determined the steering direction of the robot. A logical next step is to incorporate information about the objects' distance and use relative motion cues to steer the robot. We already implemented the sEMD model on the digital neuromorphic processor SpiNNaker (Furber, Galluppi, *et al.,* 2014) (see Figure 7.3) and obtained first promising real-time sEMD array responses (data not shown). Additionally, it would be ideal to control the robot's steering with higher resolution than previous approaches. To this end, controlling the motors using pulse-frequency modulation (Perez-Peña, Leñero-Bardallo, *et al.,* 2017), rather than a pulse-width modulation, would allow the network to directly control the robot's behaviour without the need for time-averaging its output. Both real-time sEMD array implementation on SpiNNaker and motor control, are subject to ongoing research.

Since mixed-signal neuromorphic sensory-processing systems can by some means interact with their environment (Denk *et al.,* 2013; Milde, Dietmüller, *et al.,* 2017) and were shown to scale-up nicely, while maintaining a low power consumption and a small size, these systems represent an interesting alternative to conventional frame-based solutions (Hwu, Isbell, *et al.,* 2017; Hwu, Krichmar, *et al.,* 2017). Furthermore, in the context of search and rescue operations small and autonomous robotic agents are desired, since operation and processing time, as well as size are three crucial constraints (Calisi *et al.,* 2007; Ko & Lau, 2009).

Sensory-domain generalization

The sEMD responds to information inherent in the temporal difference of incoming stimuli. Such an encoding of Δt also appears in information processing in the brain, a prime example being the inter-aural time difference to localise the position of an auditory stimulus (Konishi, 1971; Finger & Liu, 2011).
To test generalisation properties of the presented circuit, we connected the two inputs of the sEMD circuit, namely ASE and DPI, to a Dynamic Audio Sensor (Chan *et al.,* 2007; Liu, van Schaik, *et al.,* 2010; Liu, Mesgarani, *et al.,* 2010). The time difference in incoming spikes originates from the two microphones on the audio sensor. After tuning the biases to fit the dynamic range of incoming stimuli ($10\ ns\ < \Delta t < 700\ \mu s$) the sEMD could encode the input Δt into a burst of spikes and furthermore, the sEMD activity was used to extract the position of a sound source using a network implemented on a SpiNNaker (Furber, Galluppi, *et al.,* 2014) board (data not shown).

Biological plausibility

Our sEMD model exhibits performances (see Figure 3.9) similar to those of the well-established frame-based Elementary Motion Detector (EMD) model proposed by Hassentstein and Reichardt more than 60 years ago (Hassentstein & Reichardt, 1956). It has been recently shown by Borst and colleagues that many of the original predictions by Hassentstein and Reichardt were found to be valid and large parts of the neural correlates of the EMD model have been identified in vivo (Maisak *et al.,* 2013; Mauss *et al.,* 2014; Borst & Helmstaedter, 2015). The original EMD model was used to explain motion detection mechanisms in the insect's visual pathway, which at the level of small local neurons in the peripheral visual system, as in the vertebrates retina, relies to large extend on graded potentials to convey information. Whilst the classical EMD model explains how to estimate local motion based on graded potentials, it fails to explain how to extract motion from spiking inputs. In contrast, the sEMD model might provide insights in how to extract motion information if the underlying neural code relies on spikes, as are assumed to play a dominant role in the mammalian visual cortex. Here, we presented a biologically plausible solution of how such a computation could be performed using precise spike-timing.

Postamble

We were able to show that by adapting the synaptic efficacy based on the relative timing of spikes, SNNs are capable of extracting relative distance information. This information can be used to scale avoidance behaviour depending on how close the agent is to an object. The Adaptive Synaptic Efficacy (ASE) mechanism constitute the core principle to do so and can be simulated in software and emulated with mixed-signal sub-threshold CMOS circuit. The neuronal algorithm performs a precise temporal correlation analysis of pairs of spikes. The algorithm and its circuit implementation are agnostic to the source of the spikes; it only cares about their relative timing and expresses this measurement in a burst of spikes, in which the temporal information of the stimulus is preserved within its output rate, but also within the ISI distribution of the burst. In the study above we could demonstrate its application in an open-loop navigation scenario. In a follow-up study, we also demonstrated that the proposed neural algorithm is capable of avoiding collisions when the estimated collision avoidance direction affects the agent's steering behaviour in a closed-loop (see Figure 3.10). As we were able to show the proposed mechanism is capable of extracting relative distance information (see Figure 3.8), however, the underlying computation, i.e. temporal correlation detection, is of much more universal relevance. By scaling the EPSC based on the relative timing, the downstream neuron, which integrates the incoming EPSC emits a burst of spikes whose ISI distribution preserves information about the temporal correlation, i.e. the Δt at the input (see Figure 3.7). What is interesting about this scaling mechanism is that it is independent of the associated synaptic weight. This separation of synaptic weight and transmission efficacy implies that if we want to adjust the synaptic weight to extract spatio-temporally correlated event or spike patterns, we could dynamically scale how strongly the presence of a given learned pattern affects downstream processing. In other words, the adaptive scaling, i.e. gain modulation, can potentially be used as either a predictive or an attentive signal, down- or up-regulating information transmission without affecting the learned synaptic weights. This represents a novel way of gating information in spiking neural networks, which mainly depends on the relative timing of spikes arriving at a particular neuron. The positive synaptic efficacy scaling could be seen as an *amplification/attention signal*, which emphasises an "expected" relative timing of spikes. The negative efficacy scaling, on the other hand, is rather like a self-generated *prediction signal*, which de-emphasises an "expected" relative

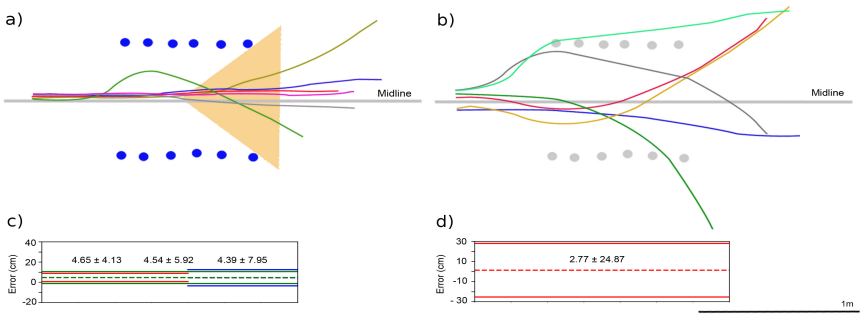


Figure 3.10: Closed-loop trajectories of the robot with and without a corridor. In the presence of a corridor, the robot successfully performs centering behaviour. The proposed sEMD was simulated on the SpiNNaker platform, which was mounted on top of the robot together with an eDVS. The SpiNNaker board simulated in total 4096 sEMDs. a) Six trajectories with a corridor and with the yellow viewing angle of the eDVS are shown. b) Trajectories without any objects in the room are demonstrated. c,d) Average trajectory with calculated standard deviation across different runs. The starting point in the setup without corridor is always normalized to zero for every run. For the trials with the corridor, the mean error calculation taken up to the point where the robot cannot see any obstacles anymore. In the presence of obstacles, which form a corridor (c), the error is, as expected, smaller compared to the open environment control experiment (d). Thus, demonstrating successful centering behaviour as long as objects are within the field of view.

timing of spikes. One interesting aspect to study in this context is the effect of temporal noise or jitter and how it affects the gating of information, and to what extent the synaptic efficacy scaling mechanism is robust to temporal noise. Temporal noise, i.e. uncertainty, may actually be beneficial, since no prediction, top-down attention, or bottom-up prediction, will be perfect in time. Hence, the ASE mechanism will never set the synaptic efficacy to the highest possible value (an avalanche of spikes), nor set the efficacy to the lowest possible value (killing the activity of the network completely).

In the next part of the thesis, we will investigate how the ASE mechanism can be combined with synaptic plasticity to extract spatio-temporally correlated activity. The following investigations will help us to understand how a SNN can self-generate temporal predictions as a form of teacher signal and how synaptic plasticity can co-exist with synaptic gain modulation. But most importantly the following part tries to formulate a neural algorithm which is able to adjust its synaptic parameters continuously and incorporates a temporal representation

of time in ongoing processing. The following part investigates how time could serve as a binding entity between what the system learned and what it is capable of explaining/predicting, which represents a significant extension of scene understanding in SNNs.

PART III

Scene understanding

4

Online Clustering of Temporal Activity

Preamble

In the last two chapters we demonstrated two neurally implemented systems that, by generating appropriate collision avoidance behaviour given visual input, were able to behave autonomously in their visual environments or scenes (see Chapter 2 & 3). The system proposed in Chapter 3 generated appropriate collision avoidance responses by extracting optic flow, to determine its avoidance response, whereas the system proposed in Chapter 2 low-pass filtered its inputs and determined its avoidance response solely based the relative retinal location of an obstacle. Both approaches, however, hard-coded the way information was extracted from the sensory input stream and were characterised by a hard-coded "decision tree", i.e. which behaviour to execute given the network's activity. The networks were not able to adjust their synaptic weights, so they needed manual parameter tuning. Furthermore, device mismatch present in both systems could only be compensated by population averaging and not be used for computation. Neither of the presented approaches to visual scene understanding extracted spatio-temporal features beyond optic flow, even though the proposed Adaptive Synaptic Efficacy (ASE) mechanism (see Sections 3.2.1 & 3.2.2) could demonstrate the capability to extract and preserve the temporal structure from its inputs (see Section 3.2.2 & Figure 3.7).

To expand the system's capability to understand visual scenes without the need to hard-code the feature extraction stage, the system should incorporate some sort of learning. If the incorporated learning rules are unsupervised the system

is capable of allocating its computational resources, e.g. neuron and synapses, in a flexible, data-driven manner. A sensible learning objective in the context of visual scene understanding is to to be able to self-generate activity to predict future inputs. Feedback can either be provided by processing elements located at higher stages in a processing hierarchy or be provided by lateral processing elements in the same processing level.

A circuit motif which can be used to study lateral feedback is networks of locally interconnected excitatory-inhibitory neurons which can be found throughout the mammalian neocortex, e.g. in layer 2/3, layer 4 and layer 6 (Binzegger *et al.,* 2004; Grill-Spector & Malach, 2004). These simple networks are already capable of producing rich dynamics including signal restoration, signal amplification (Indiveri, 1997; Hahnloser, Douglas, *et al.,* 1999; Indiveri, Oswald, *et al.,* 2002; Kurt *et al.,* 2008) or even decision making (Hahnloser, Sarpeshkar, *et al.,* 2000; Horiuchi, 2009; Pfeiffer *et al.,* 2010; Mostafa, Müller, *et al.,* 2013). For more details refer to (Maass, 2000; Oster *et al.,* 2009, & Section 2.2.4). To achieve the dynamics, excitatory-inhibitory neurons typically show local excitatory connections and widespread inhibition. This configuration is referred to as winner-takes-all (WTA). We demonstrated in Chapters 2 & 3 that the connectivity scheme of WTAs is suited to perform a collision avoidance decision based on noisy sensory data. Lateral feedback projections are usually initialised according to a Gabor-like connectivity kernel and kept fixed thereafter. In the case of WTAs the lateral excitatory projections can be understood as a hard-coded transition probability, in which state and space are entangled because of the nearest neighbour connectivity (Kappel *et al.,* 2014). However, anatomical finding suggest that the connectivity kernel, which is needed in a WTA to perform stable decision making is actually flipped, i.e. inhibitory interneurons project more locally compared to their excitatory partners (Lund, 1987; Douglas & Martin, 2004, 2009; Muir & Cook, 2014).

The local excitatory-inhibitory networks described before are highly interconnected with one another and form together the canonical microcircuit which describes the anatomical organisation of a single cortical column (Douglas, Martin & Whitteridge, 1989, & see Section 4.2.1) To better understand the computational power of simple excitatory-inhibitory networks embedded in a meta-architecture inspired by the canonical microcircuit we (i) make no a priori assumption about the connectivity by randomly initialising the synaptic projections in an all-to-all fashion and (ii) we equip synaptic projections with local, event-driven plasticity rules. The developed meta-architecture represents a modular,

scalable building block for SNNs. This building block is capable of extracting spatio-temporally correlated features from a stream of asynchronous events. It represents a first proof of concept to understand the computational role of lateral and feedback recurrent projections for online and time-continuous processing.

4.1 Introduction

BIOLOGICAL neural networks, such as the highly recurrent network motifs found in the neocortex, are thought to be the hallmark of computation. In contrast to synchronous, clock-driven computing architectures is the computation in biological computing architectures, i.e. neural networks, rather asynchronous and event-driven. As soon as an action potential arrives at a synapse, it triggers an Excitatory Post-Synaptic Current (EPSC) or Inhibitory Post-Synaptic Current (IPSC), which follows, to first approximation, the temporal dynamics of a first-order low-pass filter (Dayan & Abbott, 2001). The height of the EPSC is determined by the associated synaptic weight, and the temporal decay is determined by the synaptic time constant. The post-synaptic neuron, in turn, integrates the synaptic current which propagates along the dendrite into its soma and initialises a spike at the distal end of the axon initial segment less than a millisecond after its membrane potential crosses its threshold potential (Kole *et al.,* 2008). The integration is affected by neuronal time constants which are determined by a myriad of voltage-gated ion channels located in the membrane (Hodgkin & Huxley, 1952; Soldado-Magraner *et al.,* 2017). Moreover, neuronal population and their associated cortical areas are topographically organized, suggesting that neural computation in the neocortex is carried out in parallel (Hubel & Wiesel, 1962; Hubel & Wiesel, 1965, 1977; Essen & Zeki, 1978; Van Essen *et al.,* 1992; Grill-Spector & Malach, 2004; Patel *et al.,* 2014) and anatomical findings suggest that a substantial amount of computation takes place in local ensembles (Ercsey-Ravasz *et al.,* 2013). The cortical organisation principles and the properties of their processing elements to represent time, at least on a short time scale ($\leq$ 100 ms), makes cortical networks an interesting model to study network dynamics in the context of time-continuous inputs and representation of time in ongoing processing.

Event-based sensory systems can provide such time-continuous inputs to biologically inspired networks of spiking neurons (for more details on event-based sensing see Section 1.1). Event-based sensors do not sample the environment synchronously across the sensor array at fixed intervals but provide an asynchronous, data-driven and fully parallel output. Event-driven neuromorphic processors provide an ideal computing framework to operate on such a time-continuous stream of asynchronous events (for more details see Section 1.2). The implemented network topology defines the algorithms executed on these neuromorphic processors. A major strength of these processors is that they nat-

urally provide a distributed computing framework which at least in some cases is completely asynchronous. Since both event-based processing architectures and event-based sensors are able to let time represent itself, the underlying event-based algorithms should, in principle, be able to exploit this temporal representation.

The present work shows how event-driven, local plasticity mechanism for excitatory and inhibitory synaptic projections enable a recurrent SNN to learn spatio-temporal prevalent patterns. The recurrent SNN performs an Online Clustering of Temporal Activity, which inspired the name for the proposed network architecture: OCTA. We will demonstrate in detail how Spike-Time Dependent Plasticity (STDP) in feedforward and recurrent excitatory synapses, when combined with a novel Activity Dependent Plasticity (ADP) rule for inhibitory synapses, leads to the formation of unique spatio-temporal receptive fields and an implicit temporal representation of the provided time-continuous inputs. The implicit temporal representation of time enables a single Online Clustering of Temporal Activity (OCTA) module to perform temporal predictions of its input by adaptively scaling synaptic transmission efficacy based on temporal correlations of pairs of spikes. We will further demonstrate that the proposed OCTA module not only can deal with noise present at its inputs but also with noise present in its processing elements. Moreover, we will show that a single OCTA module can at least partially be mapped to mixed-signal, sub-threshold neuromorphic processing hardware.

4.1.1 Scene understanding in neural-based systems

Conventional approaches to visual scene understanding either rely on single frames or rely on successive frames of a video (see Section 1.3). While some of the discussed approaches rely on annotated training data, autoencoders overcome this limitation as the desired output of autoencoders is to generate the presented input (Hinton & Zemel, 1994; Vincent, Larochelle, Bengio, *et al.,* 2008; Vincent, Larochelle, Lajoie, *et al.,* 2010, & see Section 1.3 for more details).

Burbank (2015) proposed for the first time an autoencoder scheme for spiking neurons by formulating STDP rule which is capable of mirroring the encoding weights, such that $\mathrm{W}_{encoding} = \mathrm{W}^{\mathrm{T}}_{decoding}$. However, to achieve symmetric weight changes between the encoding and the decoding weight matrix Burbank used the same relative timing for both weight matrices in the STDP update (Burbank, 2015). In other words, Burbank replaced the temporal relation of pre and post

synaptic spiking ($\Delta t = t_{pre} - t_{post}$) with a temporal relation of visible layer, i.e. input layer, and hidden layer, i.e. embedding layer, spiking ($\Delta t = t_{visible} - t_{hidden}$). This is indeed a nice mathematical trick to fulfil Equation 1.2, however, poses a biologically implausible constraint for strictly local weight updates. Additionally, static images were presented to the spiking autoencoder preventing the autoencoder from operating in the temporal domain.
Here we extend autoencoders to be able to operate in the temporal domain. This extension requires the autoencoder to have an implicit temporal representation of time in its ongoing processing, but more importantly, it allows the autoencoder to work with asynchronous events as emitted by a DVS naturally. The proposed solution could be a general solution to event-based feature extraction, as in fact any event-stream which carries an additional payload (see Section 1.1) can be seen as a payload-temporal pattern, whether the payload is pixel coordinates, frequency band or any other feature. Before we sketch such a temporal autoencoder, we first highlight the drawbacks of previously proposed unsupervised feature learning algorithms designed explicitly for asynchronous event-based data.

4.1.2 Unsupervised feature learning approaches for event-based data

An important step to incorporate an implicit representation of time in ongoing processing and feature extraction is to replace static frames at the input which are sampled synchronously at fixed frame-rates with event-based asynchronously sampled, time-continuous inputs (for more details see Sections 1.1 & Liu, Delbruck, *et al.,* 2014). Event-based approaches to feature extraction have been proposed which try to find matching spatio-temporal feature templates (see Section 1.4 for more details). While some approaches calculated the distance between a detected time surface and feature templates explicitly (Lagorce, Orchard, *et al.,* 2017; Afshar *et al.,* 2019), others compute this distance implicitly by using a STDP rule (Bichler, Querlioz, *et al.,* 2012). These approaches use lateral inhibitory feedback to install competition among feature detectors, thus guaranteeing to learn unique features.
Lagorce, Ieng, *et al.* (2015) proposed an interesting alternative feature extraction algorithm which uses recurrently connected echo-state networks (ESNs) (Jaeger, 2002) in combination with lateral inhibitory feedback. The ESNs enable the proposed algorithm to classify spatio-temporal patterns based on their spatio-temporal sequence. The communication between feature detectors served, as

the other approaches mentioned above, only to introduce competition.
Lateral and top-down excitatory and inhibitory feedback, however, was shown to convey beneficial information for incorporating spatio-temporal expectations in experimental findings (e.g. Mauk & Buonomano, 2004; Leinweber *et al.,* 2017; Keller & Mrsic-Flogel, 2018) and in theoretical considerations (e.g. Jaeger, 2002; Vogels & Abbott, 2009; Kappel *et al.,* 2014; Deneve & Machens, 2016; Stroud *et al.,* 2018). To utilize the recurrent excitatory and inhibitory connections to encode additional information about the perceived input and help the network to incorporate spatio-temporal expectations the feedback projections need to be subject to learning. By adjusting these synaptic projections, the network will have an implicit temporal representation of time in its ongoing processing.

In this chapter we propose a meta-architecture for recurrent SNNs called OCTA which is capable of learning and predicting spatio-temporally correlated patterns of events by adjusting not only feedforward but also excitatory and inhibitory recurrent synaptic projections. The predictive fields are learned locally based on self-generated activity which is compared to future inputs. The resulting difference between self-generated activity and future inputs is used to gate information transmission in an OCTA module, i.e. do not transmit information which can be predicted by the network's dynamics already.
While a local unsupervised plasticity rule such as STDP can detect correlated activity across time and space and can adapt the respective connections to become not only correlated but also causal, the role of inhibition provided by the inhibitory interneurons is of a different nature. Firstly, inhibition installs competition among excitatory neurons, meaning that activity is temporally sparsified. Secondly, inhibition in combination with STDP enables the excitatory neurons to become selective to unique input patterns. Thirdly, inhibition acts as a homeostatic mechanism, which tries to balance excitation and thus drives the network into an efficient operation regime.
In short, we propose a building block for spiking recurrent neural networks that is:

- Spike-based (works with the event-based inputs)
- Robust to noise (can be emulated on neuromorphic hardware)
- Unsupervised (does not need labelled data)
- Scalable (estimates errors locally)
- Learning online (adapts to changing environments)

4.2 Methods

All software simulations conducted within this study were done using the software toolbox to simulate spiking neural networks called Brian2 (Goodman & Brette, 2009). Additionally, we developed a software toolbox, called teili, which provides user-friendly, high-level interfaces to construct neurally implemented algorithms, such as WTA or reservoir networks. The toolbox enables the user to test the effect of synaptic plasticity mechanisms quickly, e.g. STDP or novel developed learning rules, in the aforementioned neural algorithms by formulating each algorithm as a modular building block.[1].

4.2.1 The canonical building block

It is, however, not only the binary connectivity profile of the network which describes the incorporated computations, but also the associated synaptic weight and time constant. Douglas, Martin and colleagues performed thorough in vivo intra-cellular recordings of pyramidal neurons in the cat's visual cortex (Douglas, Martin & Whitteridge, 1989; Douglas & Martin, 1992a). They could not only identify stereotypical response profiles of pyramidal neurons upon visual stimulation but using anatomical and functional analysis they could report a re-occurring, stereotypical circuit motif - the canonical microcircuit (Douglas & Martin, 1992a; Douglas & Martin, 1992b,c; Binzegger *et al.,* 2004; Douglas & Martin, 2004; Douglas & Martin, 2007a; Douglas & Martin, 2007b). The originally proposed canonical microcircuit depicted in Figure 4.1b represents a major abstraction from anatomical reality. Without this abstraction we would fail to find this canonical processing and organisation scheme, as neurons tend to show a quite diverse and inhomogeneous response profile (Masland, 2004; Soldado-Magraner *et al.,* 2017), morphology (Markram, Toledo-Rodriguez, *et al.,* 2004; Oberlaender *et al.,* 2011) and diverse computational function (Markram, Toledo-Rodriguez, *et al.,* 2004; Kepecs & Fishell, 2014). Secondly, within a single "cortical column" one can find at least one connection between any two neuronal sub-populations (Binzegger *et al.,* 2004; Grill-Spector & Malach, 2004; Jiang *et al.,* 2015). If we, however "zoom out", and ignore connections with small absolute numbers of synapses, a repeating, almost stereotypical connectivity scheme reveals itself (Binzegger *et al.,* 2004, and see Figure 4.1). The canonical connectivity scheme represents a generally found connectivity rule

[1] Teili is available online: https://code.ini.uzh.ch/ncs/teili/ and its documentation can be found here: https://teili.readthedocs.io/en/latest/

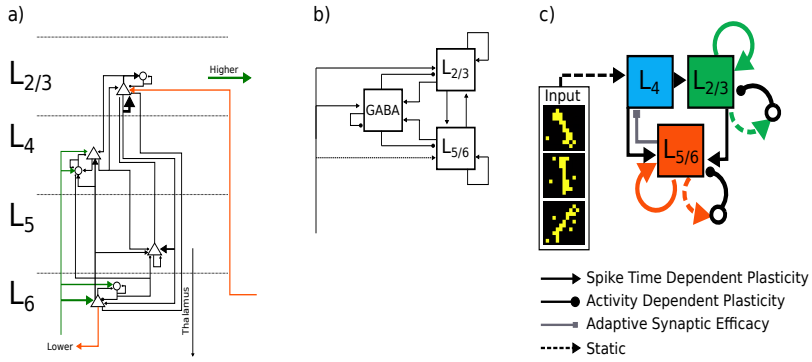


Figure 4.1: Cortical organization principles and the Online Clustering of Temporal Activity module. a) Abstracted cortical connectivity within a cortical column. Triangles represent excitatory pyramidal cells, and open circles represent inhibitory interneurons. Arrows indicate excitatory synapse, whereas dots represent inhibitory synapses. The line thickness represents the number of counted synaptic connection. Data taken from (Binzegger *et al.,* 2004), connections with less than 200 synapses were excluded for clarity. b) The originally proposed canonical microcircuit. The figure is redrawn from (Douglas, Martin & Whitteridge, 1989). c) A single Online Clustering of Temporal Activity module. The asynchronous input pattern is provided to L_4, which in turn projects to the recurrently connected excitatory-inhibitory neuronal populations $L_{2/3}$ and $L_{5/6}$. $L_{2/3}$ also projects into $L_{5/6}$. The input is connected to L_4 with one-to-one synapses. The excitatory neurons project random-fixed synapses to their respective inhibitory interneurons connectivity probability $p = 0.5$, while the interneurons project recurrently to themselves with a connectivity probability $p = 0.1$. All plastic synaptic projections are all-to-all.

of neocortical organization. The mammalian neocortex is characterised by a six-layered structure, in which thalamic or lower level synapses project primarily to excitatory pyramidal cells and inhibitory interneurons of layer 4 (L_4) and layer 6 (L_6) (see Figure 4.1a). L_4 pyramidal cells in turn project to neurons in layer 2/3 ($L_{2/3}$) and layer 5 (L_5). Neurons in L_6, on the other hand, project up to neurons located in L_4 and L_5. $L_{2/3}$ shows massive recurrent projections, has axonal projections into L_5 and L_6 and provides the input to higher level L_4 neuronal populations. While L_5 is known to be the cortical output layer towards subcortical structures, such as superior colliculus, the cerebellum and the spine, L_6 projects intra-cortically to lower-level $L_{2/3}$ and primary and higher-order thalamic nuclei. Douglas, Martin & Whitteridge (1989) originally proposed the canonical organization of the neocortex and used computer simulations to reproduce neuronal dynamics recorded in-vivo. Douglas and colleagues claimed

in their study (i) that excitation and inhibition cannot be viewed as separate events, (ii) that the major excitatory drive of cortical processing originates within cortex and is not provided by thalamic inputs and (iii) that the observed temporal dynamics ($\leq$ 200 ms) extend well beyond the dynamics of synaptic time constants ($\approx$ 20-50 ms), thus suggesting that cortical processing cannot rely on precise timing of synaptic events (Douglas, Martin & Whitteridge, 1989). While excitatory-inhibitory balance has been shown to be beneficial for stable neural processing (Vogels, Sprekeler, *et al.,* 2011; Deneve & Machens, 2016), and recurrently connected neural populations encode temporal expectations (Kappel *et al.,* 2014; Stroud *et al.,* 2018) and can perform state-dependent computations (Rutishauser & Douglas, 2009; Neftci, Binas, *et al.,* 2012), the third claim of Douglas, Martin & Whitteridge (1989) should be viewed with caution. The model used to reproduce in-vivo dynamics modelled each layer as an average single cell response, whose output represented the average layer activity (Douglas, Martin & Whitteridge, 1989). When spatio-temporally averaging cell activity, it is quite likely that precise fluctuations will perish. Secondly, the model described a single cortical column(Douglas, Martin & Whitteridge, 1989), meaning that top-down and lateral feedback were not incorporated (Binzegger *et al.,* 2004; Cudeiro & Sillito, 2006; Leinweber *et al.,* 2017; Keller & Mrsic-Flogel, 2018). And thirdly, the recordings were performed in anesthetised and paralysed animals, which might have introduced a non-negligible bias (e.g. see Goltstein *et al.,* 2015).
An alternative explanation for the much longer observed neuronal dynamics compared to the synaptic time constants is that each recurrent loop represents a signal which is delayed by at least one synaptic time constant. If we think of the cortex as recurrently connected neurons in a recurrently connected column in a recurrently connected hierarchy of columns, much longer temporal dynamics could emerge through inputs originating from higher-level hierarchical neurons[2]. To investigate this alternative explanation of the emergence of longer time evolution, we propose as a first step a recurrent SNN, named OCTA, which is subject to precisely timed input events, with the objective of online clustering of temporal activity and performing temporally precise predictions.

[2]This statement is valid as long as feedback projections are non-random. For randomly connected neural networks the longest temporal memory is determined by the longest synaptic time constant present in the network (Lukoševičius & Jaeger, 2009).

OCTA module connectivity

The basic Online Clustering of Temporal Activity (OCTA) module is depicted in Figure 4.1, c. A single OCTA module comprises of three neuronal populations which we named according to the different cortical layers (see above). The input is sent to the neuron population named L_4, which in turn is connected via excitatory plastic synapses to neuronal populations named $L_{2/3}$ and $L_{5/6}$. Both populations, i.e. $L_{2/3}$ and $L_{5/6}$, are recurrently connected populations with a pool of inhibitory interneurons. The recurrent excitatory synapses, like the feedforward synapses from L_4, are subject to STDP, whereas the synapses from the excitatory pool into their respective inhibitory interneuron pool are random and fixed with connectivity probability $p = 0.5$[3]. The synapses projecting from the pool of inhibitory interneurons are subject to learning with the objective of balancing excitatory activity over time by introducing competition which leads to unique receptive fields (for more details see Sections 4.2.3 & 4.3).

4.2.2 Neuron and synapse models

A single OCTA module used in this study consists of 286 adaptive, exponential, leaky integrate and fire neurons (Indiveri, Chicca, *et al.,* 2006; Indiveri, Linares-Barranco, *et al.,* 2011), of which 249 are excitatory, and 37 are inhibitory interneurons[4]. The network is initialised with 45300 plastic and 1350 non-plastic synapses (see Figure 4.1, c). The neuronal dynamics are described as follows:

$$\tau \frac{dI_{mem}}{d_t} = \frac{\frac{I_{th}}{I_\tau}\left(I_{in} + I_a - I_{shunt} - I_{ahp}\right) - I_{th} - \left(1 + \frac{I_{shunt} + I_{ahp} - I_a}{I_\tau} I_{mem}\right)}{\frac{I_{th}}{(I_{mem} + I_o)} + 1}, \tag{4.1}$$

where I_{mem} describes the membrane potential, I_{in} is the sum of all dendritic input currents, I_{shunt} is the somatic inhibitory current, I_{ahp} is the after hyperpolarisation current, which emulates the spike frequency adaptation mechanism, I_{th} describes the input conductance and I_a models the exponential gain factor. I_τ represents the leakage current and is related to the somatic time constant τ as follows:

$$\tau = \frac{C_{mem}\, U_t}{\kappa\, I_\tau}, \tag{4.2}$$

[3] If the probability of connection between excitatory and inhibitory neurons is increased, the two neuron populations will start to oscillate at a temporal scale solely determined by the synaptic time constant. An example of the oscillatory behaviour can be seen upon initialisation of the network (see Figure 4.4a).

[4] Note that these numbers were chosen because of the used input dimensionality not because of inherent limitation of the proposed OCTA module.

where C_{mem} represents the membrane capacitance, U_t is the thermal voltage, and κ is the sub-threshold slope factor. Although the neuronal membrane potential of neurons is the voltage between the cell interior and exterior and although synaptic dynamics describe the development of ion-channel conductances over time, our mixed-signal sub-threshold neuromorphic circuits use currents to represent these measurements and can therefore be said to be operated in current-mode. Equation 4.1 describes the current-based neuron model as first presented in (Indiveri, Chicca, *et al.,* 2006) and implemented on the DYNAP (Moradi *et al.,* 2018). The synaptic dynamics of both plastic and non-plastic synapses are described as follows:

$$\tau \frac{dI_{syn}}{dt} = \frac{-I_{syn} - I_{gain}}{\frac{I_{gain}}{I_{syn}} + 1}, \tag{4.3}$$

where I_{syn} represent the post-synaptic current (PSC), I_{gain} is modelling the synaptic gain and has a multiplicative effect on top of the synaptic weight, and τ represents the synaptic time constant. As all incoming currents are summed by the post-synaptic neuron, the synaptic type, i.e. excitatory or inhibitory, is determined by the sign of I_{syn}, which in turn is set by the weight associated to the given synapse. On the arrival of a pre-synaptic spike the synaptic variable I_{syn} is updated as follows:

$$I_{syn} = I_{syn} + \frac{I_w \, w \, I_{gain}}{I_{tau} \, \frac{I_{gain}}{I_{syn}}}, \tag{4.4}$$

in which I_w is the current flowing through transistor M_5, which is scaled by w (see Figure 3.2 for circuit schematics). For non-plastic synapses w is 1, whereas in the case of plastic synapses w is updated according to Eq. 4.5 and ranges between $[0, 1]$ (see Section 4.2.3 for more details). The plastic weights were sampled from a γ-distribution with $k = 0.4$ and $\Sigma = 0.9$ if not stated otherwise. The learned receptive fields when weights were initialised from on a γ-distribution showed sensible orientation preference in all conducted experiments and also showed stable orientation preference when the network was simulated longer. We performed initial tests using a Gaussian-distribution to sample the synaptic weights with $\mu = 0.45$ and $\sigma = 0.25$, however, the network failed to form stable receptive fields over time in more than 90% of the cases. In the cases which yielded sensible receptive fields, the weight distribution converted through the process of learning and due to clipping of values outside the range of $[0, 1]$ into a bimodal distribution similar to a clipped γ-distribution. Hence, we decided

to use a γ-distribution rather than a Gaussian-distribution in our simulations to initialise the synaptic weights.

4.2.3 Plasticity mechanism

Spike-Time Dependent Plasticity

The synapses follow the well-known DPI synapse dynamics as governed by Eq. 4.3. To incorporate spike-based learning, i.e. STDP, in the plastic excitatory synapses (see Figure 4.1c) we add two additional terms to model the weight update based on the relative timing of the pre- and post-synaptic spiking (see Eq. 4.5, as proposed by (Song *et al.,* 2000)).

$$\Delta w = \begin{cases} a^+ \cdot exp^{\left(\frac{t_{pre}-t_{post}}{\tau^+}\right)}, \; if \; t_{pre} \leq t_{post} \\ a^- \cdot exp^{\left(\frac{t_{pre}-t_{post}}{\tau^-}\right)}, \; if \; t_{pre} > t_{post}, \end{cases} \tag{4.5}$$

where $+$ and $-$ represents the weight increment and decrement respectively, t_{pre} and t_{post} are the time when the pre and post synaptic neuron elicited a spike and τ describes the temporal window in which the synaptic weight is effectively updated. Furthermore, synapses which project from the pool of inhibitory interneurons into the excitatory neurons (see Figure 4.1c) follow an Activity Dependent Plasticity rule (see below for more details).

Activity Dependent Plasticity

The role of inhibition in neuronal processing is subject to debate and how inhibitory synaptic strength is adjusted is far away from being as clearly understood as the role of learning in its excitatory counterpart (Kullmann *et al.,* 2012; Hangya *et al.,* 2014). Inhibitory interneurons are thought to regulate excitatory activity by balancing their activity (Brunel, 2000; Wehr & Zador, 2003; Vogels & Abbott, 2009; Vogels, Sprekeler, *et al.,* 2011; Deneve & Machens, 2016). Furthermore, inhibition introduces competition among excitatory neurons as inhibitory projections tend to target local ensembles of excitatory neurons (Lund, 1987; Douglas & Martin, 2004, 2009; Muir & Cook, 2014). As our functional understanding of inhibitory interneurons during computation is limited, models often incorporate random, fixed synaptic projections which are not subject to any plasticity. To gain a better understanding of the computational role of inhibition in the context of event-based processing we propose an inhibitory plasticity rule which considers sub-threshold fluctuations of the post-synaptic membrane

potential. The proposed Activity Dependent Plasticity (ADP) rule adjusts the associated synaptic weight to introduce competition among the excitatory neurons and to balance the excitatory activity, based on the post-synaptic neuron activity σ_i. We calculate the post-synaptic neuron activity as follows:

$$\sigma_i = \int_{t_0-\Delta t}^{t_0} \mathrm{I}_{mem\ i}(t)\, e^{\frac{(t-t_0)}{\tau}} dt, \tag{4.6}$$

where σ_i is the integrated post-synaptic membrane current I_{mem}, t_0 is the time of the inhibitory spike and Δt denotes the time window. The measured membrane current is convolved with an exponentially decaying kernel with a decay time constant τ. The resulting σ_i is normalised to have values between 0 and 1. Every time an inhibitory neuron spikes, the normalised post-synaptic activity $\hat{\sigma}_i$ is compared to a threshold σ_{target}, and the weight is updated as follows:

$$\Delta w = \eta \cdot (\hat{\sigma}_i - \sigma_{target}), \tag{4.7}$$

where η is the learning rate which is set to 0.001. The thresholds (see Eq. 4.7) are synapse specific and are randomly initialised from a uniform distribution in a range between 0.3 and 0.5. This learning rule is similar to (Vogels, Sprekeler, *et al.,* 2011), but instead of looking at the output firing rate we integrate the exponentially weighted somatic current I_{mem} within a sliding window. Hence, the novel plasticity rule proposed here does reflect sub-threshold fluctuations of the membrane potential and thus reflects inputs which did not trigger the generation of an action potential.

4.2.4 Dynamic routing of information by synaptic efficacy modulation

The aforementioned local plasticity mechanisms allow the network to capture spatio-temporally correlated inputs while balancing the network's activity, and by introducing competition among excitatory neurons diversify the captured spatio-temporal features. If we want to design a system which uses the learned features to make temporal predictions about future neuronal activity we propose to adjust the synaptic transmission of predicted activity. We previously could show that by varying the threshold of the DPI synapse with a time varying signal, the synapse acts as a temporal correlation detector (see Section 3 & Milde, Bertrand, Ramachandran, *et al.,* 2018). The mechanism enabling the synapse to detect temporal correlation can be understood as adaptively varying the synaptic

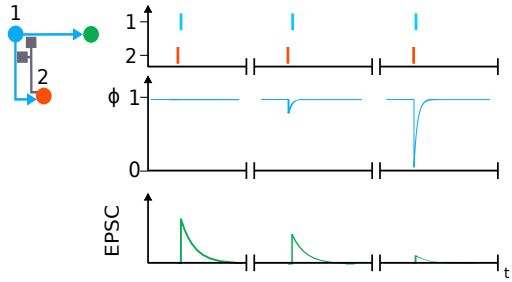


Figure 4.2: Adaptive Synaptic Efficacy (ASE) mechanism. Neuron 2 sets the synaptic gain of neuron 1. The synaptic gain value, i.e. the height of the drop in synaptic gain, is learned via STDP with a time window of 5 ms. At the time of a pre-synaptic spike in neuron 2, the synaptic efficacy of neuron 1 is lowered and exponentially decays back to its baseline value of 1 with a time constant of 1.5 ms.

efficacy, thus we named it the Adaptive Synaptic Efficacy (ASE) mechanism. In other words, the EPSC is modulated based on the relative timing of two spikes. The two main parameters which define the operation regime are two synaptic time-constants, τ_{gain} and τ_{syn} (see Figure 3.2). While in the case of optic flow detection the efficacy was up-regulated for highly temporally correlated spikes, the synaptic efficacy modulation should also work in the opposite direction, i.e. temporally correlated spikes lead to a down-regulation of the synaptic efficacy. The positive synaptic efficacy scaling could be seen as an *amplification/attention signal*, which emphasises an 'expected' relative timing of spikes. The negative efficacy scaling, on the other hand, can be understood as a self-generated *prediction signal*, which de-emphasises an 'expected' relative timing of spikes. Instead of hand-tuning the amplitude of the time varying signal ϕ at the time of the arrival of pre-synaptic spike, we use a STDP mechanism to learn it (see Eq. 4.8).

$$\Delta\phi = \begin{cases} a^+ \cdot exp^{\left(\frac{t_{pre}-t_{post}}{\tau^+}\right)}, \; if \; t_{pre} \leq t_{post} \\ a^- \cdot exp^{\left(\frac{t_{pre}-t_{post}}{\tau^-}\right)}, \; if \; t_{pre} > t_{post} \end{cases} \tag{4.8}$$

The prediction mechanism via modulating the synaptic efficacy is incorporated in the synapses projecting from L_4 into $L_{2/3}$ and $L_{5/6}$. However, the activity of $L_{5/6}$ determines the synaptic efficacy ϕ as follows:

$$\tau_{pred} \frac{d\phi}{dt} = (1-\phi), \tag{4.9}$$

where τ_{pred} is the time constant with which the synaptic gain value converges back to 1 (see Figure 4.2).

4.2.5 Additional mechanism for learning

Device mismatch Neuromorphic processors are subject to device mismatch (see Section 1.2 & Pelgrom *et al.,* 1989). An algorithm emulated on these processors needs to to be capable of either compensating device mismatch (see Chapter 2) or adjust its parameters during training to cope with device mismatch which is characterised by heterogeneous response profiles of its processing elements. To simulate an element-to-element variability of approximately 20% we draw each parameter from a truncated normal distribution with a standard deviation of 0.2.

Weight decay All synaptic weights which are subject learning are also subject to a constant weight decay[5]. We decay the weight by a factor proportional to the associated learning rate η, i.e. $1-(\eta/10)$. This weight decay prevents synaptic weights from becoming too correlated which would cause synchronous and oscillatory network activity. Synchronous network activity can be understood as that the network's output is independent of the input, thus indicating that no information is being encoded in the ongoing activity of the network and no computation is being performed.

Weight re-initialisation Inhibition provided by the interneurons can prevent neurons from being active. In combination with a constant weight decay, this can lead to a reduction of synaptic weights to zero. To ensure that the constant weight decay does not introduce a bias we calculate the average weight per post-synaptic neuron. If the average weight is below 0.2, we re-initialise the synaptic weights associated to this neuron. The re-initialisation guarantees that each neuron can learn a unique spatio-temporal receptive field. If all weights associated with a post-synaptic neuron are being potentiated, then this leads to even stronger synaptic weights due to the correlation-based STDP paradigm. In this case, we re-initialise the weights if the average weight is above 0.8. Example receptive fields are depicted in Figure 4.7c. In biological neural networks it has been observed that weak synapses which are characterised by physically small

[5]Constant weight decay is a common practice in computer simulations (e.g. Krogh & Hertz, 1992; Krizhevsky *et al.,* 2012). and can be also observed in biological synapses (Cui *et al.,* 2004; Abraham & Robins, 2005)

spine structures get pruned (e.g. Abraham & Robins, 2005; Mitra, Jadhav, *et al.,* 2005; Restivo *et al.,* 2009) in an activity dependent manner, however, these processes happen typically on much longer time scales such as hours, days or even months. We took these observations only as an inspiration, but we do not claim to model them realistically. For more biologically realistic models and possible implementation on neuromorphic hardware, please refer to George (2017).

4.2.6 Sorting

To better understand what the network is coding for we sort the spike raster plots according to a given weight matrix. First, we compute the pair-wise similarity of the neuron weight vectors, by calculating the Euclidean distance between them. In the second step, we construct a directed graph based on the similarity of each node and allow each node to be maximally connected to any two other nodes. The resulting graph represents a permutation, in which similar weight vectors are closer to each other compared to orthogonal ones. In the following, we will refer to this by saying the spike raster plot is sorted according to a given synaptic projection or weight matrix.

4.3 Results

We use a simple spatio-temporal pattern to study emergent network dynamics of a single OCTA module. The stimulus is a rotating bar, with 18 different angular positions and a spatial extent of 10 x 10 pixels (Figure 4.1c, input). The bar itself is encoded as a set of events, similar to what an event-based sensor would perceive if an edge is crossing its pixels. Additionally, we add 20 % noise events. The bar rotates with a constant angular velocity of 3300 $\frac{^\circ}{s}$. This particular input was chosen because it represents a cyclic, relatively simple spatio-temporal pattern which makes the interpretation of the learned receptive and predictive fields and their temporal sequence more intuitive and easier to understand. Furthermore, the simplicity of the input makes it possible to more easily asses if a single OCTA module is capable of extracting the prevalent feature of different orientated edges despite the presence of noise, and also to understand which information about the stimulus is encapsulated in which weight matrix. The spatial dimension was chosen to represent a realistic size receptive fields to later learn feature from actually event-based sensors (for an analysis on the size of

receptive fields for event-based sensors, please refer to Afshar *et al.,* 2019). The input is connected one-to-one with static synapses. These connections are only there to relay the events as generated by the input stimulus to L_4 neurons. At the beginning of the simulation all synapses are initialised all-to-all, and the respective synaptic weights are randomly sampled from a γ-distribution, except those projections marked with a dashed line (see Figure 4.1c). The synapses projecting from the pool of excitatory neurons to their respective inhibitory interneurons are randomly initialised with a connectivity probability $p = 0.5$. Additionally, inhibitory neurons project to themselves with connectivity probability $p = 0.1$. The excitatory feedforward projections from L_4 to $L_{2/3}$ and $L_{5/6}$ respectively have the objective to learn spatio-temporal features, whereas the recurrent excitatory projections should capture the temporal evolution and thus represent a temporal expectation. The excitatory synapses from $L_{2/3}$ to $L_{5/6}$, on the other hand, have the objective to encapsulate the temporal relation between the learned spatio-temporal features. The respective inhibitory projections should provide the necessary competition among the excitatory neurons and balance activity therein. A special role is assigned to the projection from $L_{5/6}$ to L_4, which should predict inputs by successfully modulating the synaptic efficacy of the post-synaptic partners. We will start by looking into the activity of $L_{2/3}$ and show that the involved synaptic projections indeed fulfil the objectives mentioned above before we look into the activity of an entire OCTA module.

4.3.1 Spatio-temporal receptive fields

In total $L_{2/3}$ comprises 49 neurons, which are arranged in a $7x7$ grid (see Figure 4.3 a). Each post-synaptic weight vector is rearranged to match the input space to visualise the $10x10$ receptive field (see Figure 4.3 top row), and the corresponding histogram of the full adjacency matrix between L_4 and $L_{2/3}$ is shown below (see Figure 4.3, middle row). To easily assess the information content encapsulated by the receptive fields of neurons in $L_{2/3}$ we calculate the cross-correlation coefficients between all neurons of $L_{2/3}$ and sort the coefficients according to the neuron id (see Figure 4.3, bottom row). Upon initialisation the receptive fields are random which can easily be seen in the sorted cross-correlation coefficients. Neurons are only highly correlated to themselves while being uncorrelated to others (see Figure 4.3 a). As the receptive fields are random in the beginning, any presented input will lead to spiking activity in $L_{2/3}$, which in turn activates the inhibitory neurons, which themselves sup-

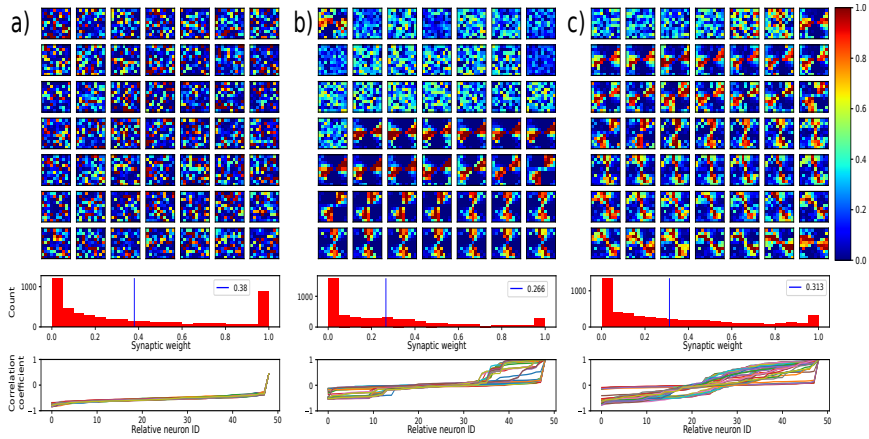


Figure 4.3: Receptive fields of excitatory neurons in $L_{2/3}$. Top row: receptive fields; middle row: histogram of adjacency matrix between L_4 and $L_{2/3}$; bottom row: sorted cross-correlation coefficients between receptive fields of neurons in $L_{2/3}$. a) Randomly initialised receptive fields before training. Weights are sampled from a γ-distribution with $k = 0.4$ and $\Sigma = 0.9$ and are clipped between 0 and 1. The average weight is 0.38. The cross-correlation analysis shows that each receptive field is only highly correlated to itself while being un- or anti-correlated to others. b) Receptive fields after 100 revolutions with random-fixed inhibition. Only 28 out of the 49 neurons were able to develop an orientation preference. However, as inhibitory weights are fixed, the learned receptive fields are highly correlated and cover only a subspace of the presented orientations. c) Receptive fields after 100 revolutions with plastic inhibition according to 4.7. 43 out of the 49 neurons were able to develop an orientation preference. The correlation analysis shows that some receptive fields are highly correlated, however, a substantial amount of receptive fields are either uncorrelated or even anti-correlated, nicely spanning the space of presented orientations.

press activity in the excitatory pool (see Figure 4.4a). The temporal evolution is only governed by the synaptic time constants and does not reflect any temporal aspects of the provided input. In the first experiment we only adjust the excitatory STDP synapses (see Eq. 4.5), while keeping the inhibitory synapses fixed ($p = 0.67$). After 10 s only 28 neurons adjusted their weights to show an orientation preference in their receptive field (see Figure 4.3b). The remaining 21 neurons have random receptive fields, which due to the constant weight decay and despite the reinitialisation have small associated synaptic weights. The receptive fields are sorted according to the learned recurrent weight matrix. Neurons with similar receptive fields show a strong similarity in their recurrent weights and thus are placed next to each other. Apart from the orientation preference, the cross-correlation analysis reveals that the learned receptive fields are highly correlated and only a few receptive fields are in fact anti-correlated (see Figure 4.3b, bottom). Even if a given neuron now tries to tune its weights to reflect a particular orientation, the static inhibitory clamp prevents it from doing so by preventing post-synaptic spiking, which is crucial for STDP. The effect of this static inhibitory clamp not only affects the formation of orientation tuning but also affects the temporal activity (see Figure 4.4b). The spike raster plot is also sorted according to the recurrent weight matrix. While the feedforward weights from L_4 to $L_{2/3}$ incorporate the spatio-temporal structure of the input, the temporal sequence is encoded in the recurrent weights, as the sorted spike raster plot shows that neurons with a particular orientation preference fire repetitively at a particular moment in time (see Figure 4.4b). In the following we will refer to this behaviour as temporal alignment. When examining Figure 4.4b more closely, however, only 28 out of the 49 feature detectors are at all active, and as the correlation analysis of the receptive fields suggests, they tend to temporally cluster due to similar receptive fields, which ultimately will lead to synchronised activity in $L_{2/3}$. The synchronisation is a natural consequence of correlation-based learning rules, such as STDP. Up to now, we kept the inhibition provided by the inhibitory interneurons static. This clearly provides the necessary competition among the excitatory feature detector neurons to learn different spatio-temporal features. However, it has the drawback that over time the receptive fields and the temporal network activity tend to synchronise, and thus do not encode any unique/useful information about the stimulus as the activity is only determined by the synaptic constants between the excitatory and inhibitory neurons and vice versa, but not by any learned weights. To overcome this limitation and allow the network to use more computational resources, i.e.

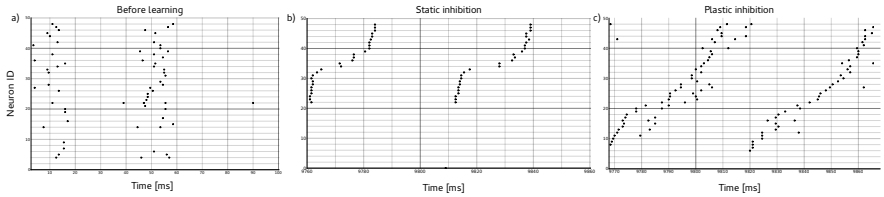


Figure 4.4: Spike raster plot, i.e. temporal activity, of $L_{2/3}$ before training and without and with plastic inhibition for a full stimulus revolution. Raster plots are sorted according to recurrent weight matrix of $L_{2/3}$. a) Randomly initialised network with randomly sorted neuron indices. No structure in the temporal activity is present. b) Temporal activity after 100 revolutions with random-fixed inhibition. Only 28 out of the 49 neurons were able to develop an orientation preference and thus are represented in the spike raster plot. While the recurrent weight matrix is able to incorporate the temporal structure of the input, the non-plastic inhibition prevents the neurons from being active during the entire stimulus presentation. Furthermore, due to the STDP rule, the network's activity starts to synchronise as static inhibition cannot prevent it. c) Temporal activity after 100 revolutions with plastic inhibition according to 4.7. 43 out of the 49 neurons were able to develop an orientation preference and thus are represented in the spike raster plot. The plastic inhibition allows the network to utilise more computational resources and thus neurons are active during the entire stimulus presentation. Plastic inhibition also prevents the network from synchronising.

neuron and synapses, we initialise the inhibitory to excitatory projection as all-to-all and make the weights subject to ADP. The inhibition, which is needed to learn different orientation tuning in the excitatory neuron pool, is adapted according to Equation 4.7, to allow each neuron to be equally active which results in more diverse receptive fields (compare Figure 4.3b & c). This increased ability to differentiate between more orientations is also reflected in the corresponding correlation analysis (see Figure 4.3c, bottom). Inhibitory weights which are too weak in the beginning lead to an over-excitation, and thus to similar orientation tuning of multiple neurons in $L_{2/3}$. This increased activity level is reflected in the normalised post-synaptic variable $\hat{\sigma}_i$ and hence these weights are strengthened according to Equation 4.7 (see Figure 4.5). On the other hand, inhibitory projections which are too strong and prevent any activity in their respective excitatory partners in $L_{2/3}$ are weakened, to allow the neuron to tune its weights to a particular orientation. By adjusting the inhibitory weights according to Eq. 4.7, 43 neurons out of the 49 learn a preferred orientation. Hence, more computational resources can be leveraged to learn and encode the presented spatio-temporal pattern. The amount of allowed activity is set by σ_{target}, which can take any

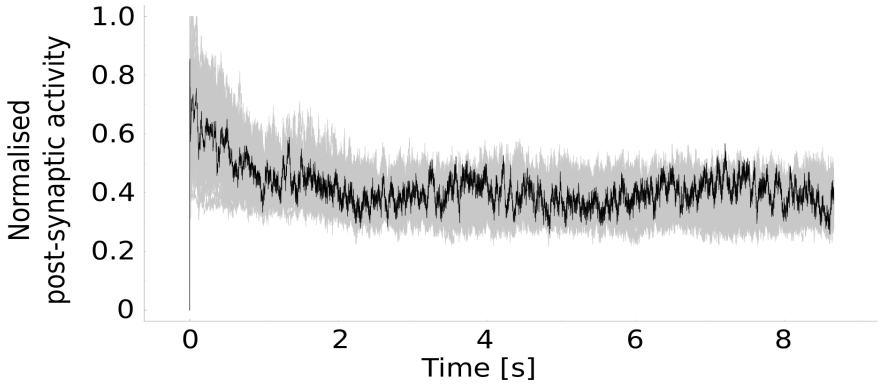


Figure 4.5: Effect of Activity Dependent Plasticity (ADP) on normalised post-synaptic activity $\hat{\sigma}_i$ of neurons in $L_{2/3}$. In total 588 traces are shown. A single normalised post-synaptic activity trace is plotted in black for better visibility. The proposed inhibitory plasticity rule is capable of adjusting the inhibitory weights, such that the difference between $\hat{\sigma}_i$ and σ_{target} is minimised. This leads not only to more neurons being recruited to tune their weights to a particular orientation, but also balances the overall activity, such that it helps to prevent oscillations from occurring.

value between 0 and 1 and is uniformly initialised between 0.3 and 0.5 for each synapse. When setting σ_{target} equal to 1, meaning no inhibitory competition is provided, the learned receptive fields are the same for each neuron, a superposition of all presented orientations. On the other hand, a σ_{target} of 0 leads to maximal inhibition, which causes the network to oscillate and learn a single preferred orientation for all neurons in $L_{2/3}$. Despite the inhibitory weight adaptation introduced by ADP, the recurrent excitatory weights capture the temporal evolution of the stimulus, and the network activity is temporally aligned (see Figure 4.4c). The inhibitory plasticity successfully balances the network activity by minimizing the difference between the post-synaptic activity $\hat{\sigma}_i$ and the respective activity target σ_{target} (see Figure 4.5). Additionally, the proposed plasticity mechanism prevents the network from synchronising, as the learned receptive field are more diverse, and each neuron has a unique temporal activity during the presentation of the stimulus.

4.3.2 Mismatch and synaptic temporal prediction

Oscillations represent a severe limitation for SNNs if the oscillatory frequency is solely determined by the synaptic time constants. The network's activity starts to oscillate either if inhibition is too strong and not subject to plasticity or if the net-

work is exposed longer than 15 seconds to the stimulus. The latter reason is due to the fact that the excitatory synapses are altered according to STDP. STDP is a temporal correlation based learning rule, which has the consequence that synaptic weights get strengthened if the respective pre and post-synaptic neurons show temporally correlated activity. Ultimately, this will cause already correlated activity to become more correlated up to the point where all weights are high. This has the consequence that excitatory neurons will emit spikes at the same point in time which in turn will trigger an avalanche of inhibitory spikes, preventing any neuron from firing until the excitatory drive is again strong enough. Hence, the network will exhibit oscillations and lose all previously encoded information as the dominant source of activity is internally generated and determined by the excitatory-inhibitory synaptic time constants.

As previously discussed, neuromorphic analogue sensory-processing systems are subject to device mismatch. The proposed meta-architecture was designed to provide an unsupervised, local feature extraction mechanism for neuromorphic sensory-processing systems. To overcome the limitation mentioned above and test if an OCTA module would be suitable for hardware emulation we added mismatch to our simulation. It turns out that the heterogeneity does not need to be compensated by population averaging as previously proposed (see Chapter 2), but in fact in combination with STDP and ADP provides the network with sufficient noise to prevent synchronous oscillations from dominating the network's activity. As in the previous simulation, the network starts from randomly initialised weights, and all plastic parameters are trained jointly. Both excitatory feedforward projections from L_4 to $L_{2/3}$ and $L_{5/6}$ respectively learn the statistically prevalent orientations of the input, which is reflected in the different receptive fields (see Figure 4.6). Note that, in contrast to previous experiments, the network is stimulated for 30 seconds (also 40, 50, 80 and 150 seconds were tested, showing similar network activity and receptive field formation). Despite the longer presentation of the input pattern the network does not start to oscillate. The combination of excitatory-inhibitory balance introduced by ADP and inhomogeneous network's parameters introduced by simulating device mismatch successfully prevented the network from oscillating.

The temporal activity of $L_{2/3}$, which is sorted according to the recurrent weight matrix shows the same temporal alignment as in previous experiments (see Figure 4.6, $L_{2/3}$ spike raster plot). The temporal activity of $L_{5/6}$ shows a similar temporal alignment, however, the spike raster plot is sorted according to the weight matrix between $L_{2/3}$ and $L_{5/6}$. This shows that the temporal information,

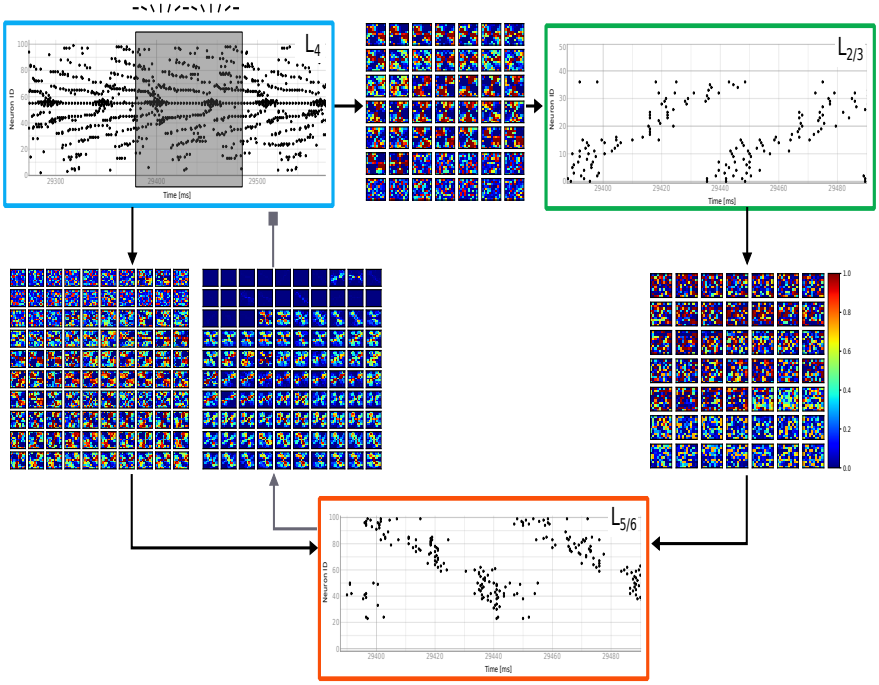


Figure 4.6: Activity of a single Online Clustering of Temporal Activity (OCTA) module. The blue box represents the temporal activity of L_4, the green box represents temporal activity of $L_{2/3}$ after sorting according to its recurrent weight matrix. The orange box represents the temporal activity of $L_{5/6}$ after sorting according to weight matrix between $L_{2/3}$ and $L_{5/6}$. All parameters of the network are subject to mismatch, i.e. show $\approx$ 20% variability. Not only is the network capable, despite the additional noise, of capturing the prevalent spatio-temporal statistics of the input, but it learns phase shifted predictive fields to temporally predict the input. Furthermore, adding mismatch allows the involved plasticity mechanisms to find a stable operating regime, without starting to oscillate and proves that the proposed meta-architecture is suited for analogue neuromorphic processing systems.

i.e. the input sequence, is preserved in the synaptic weight matrix between $L_{2/3}$ and $L_{5/6}$ (see Figure 4.6, $L_{5/6}$ spike raster plot). The only aforementioned synaptic objective we have not yet addressed is the one of the synaptic projections from $L_{5/6}$ back to its input population L_4. We hoped they would temporally predict the input. In contrast to all other synaptic projections, the predictive synapses are not learning either an excitatory or inhibitory weight, but a synaptic gain value ϕ, which upon pre-synaptic spiking lowers the post-synaptic gain of the feedforward synapses of L_4 (see Eq. 4.9 & 4.8). The synaptic gain is initialised in the beginning to one, and the respective predictive gain value ϕ is initialised to zero, thus any early activity in $L_{5/6}$ has no or minimal effect on the synaptic efficacy. The incoming receptive fields to $L_{5/6}$, as well as the outgoing predictive fields, are sorted according to the weight matrix from $L_{2/3}$ to $L_{5/6}$. While the incoming receptive fields are driven by input and respond to similar orientations as their $L_{2/3}$counterparts, the outgoing predictive fields of the same neuron tend to have a phase shifted orientation preference. The phase shift is approximately 100° (compare the receptive and predictive fields of $L_{5/6}$ in Figure 4.6). We will discuss the underlying cause of the learned phase shift between receptive and predictive fields later.

4.4 Discussion

We presented an Online Clustering of Temporal Activity (OCTA) meta-architecture for recurrent Spiking Neural Networks (SNNs), which represents a modular building block for unsupervised feature extraction suited for asynchronous event-based sensory-processing systems. The recurrent structure is inspired by the recurrent organisation principles found in neocortical columns (Douglas, Martin & Whitteridge, 1989; Binzegger *et al.,* 2004, & see Figure 4.1). We showed that a single OCTA module is capable of extracting statistically prevalent spatio-temporal features from a stream of asynchronous events, despite noise present at its sensory inputs and in its computing elements, e.g. due to device mismatch. We demonstrated that device mismatch is a feature which in combination with an inhibitory plasticity mechanism leads to stable receptive fields and balanced network activity. We further showed that the recurrent network structure enables the network to learn temporal predictions which are encoded in the predictive fields, which are tuned to the input, hence, the network has an implicit temporal representation of time in ongoing processing (see Figure 4.6).

The feedforward synapses from L_4 to $L_{2/3}$ and $L_{5/6}$ respectively incorporate

spatio-temporally correlated features which result in orientation tuning, whereas the recurrent synaptic projections incorporate the temporal sequence of the stimulus. The inhibitory projections into their excitatory counterpart provide sufficient competition while balancing the overall excitatory activity, allowing the excitatory neurons to capture the spatio-temporal evolution of the input (see Figure 4.3 & 4.4).

Recurrent weight matrix: To understand the computational role of the learned recurrent excitatory projections we calculated the Euclidean distance, i.e. similarity, of each pair of post-synaptic weight vectors. We used the similarity between each weight vector to construct a directed graph in which similar vectors or nodes are placed next to each other and with each node maximally connected to two other nodes. As each node has an index, the directed graph provides us with a permutation of indices compared to the original neuron indices. The permutation was used to sort the temporal activity covered in the spike raster plot of the neurons of $L_{2/3}$ (see Figure 4.4). This sorting procedure revealed that the temporal sequence of the input is indeed incorporated in the recurrent weight matrix, as the sorted spike raster plot showed that particular neurons fire at particular times and their corresponding receptive fields are tuned to similar orientations (see Figure 4.3 & 4.4). Due to the temporal sequence of the presented input the observed temporal alignment can be understood as expressing a temporal expectation, since neurons which show similar receptive fields are firing right after each other.

Kappel and colleagues reported previously that when recurrent excitatory weights of a soft WTA network are subject to STDP, these recurrent connections represent expectations equal to the expectation step in hidden Markov process (Kappel *et al.,* 2014), meaning that these connections are capable of learning a temporal sequence. While Kappel *et al.* (2014) showed using theoretical analysis and using computer simulations that this represents a useful feature for a variety of cognitive tasks, we could demonstrate that their findings are also valid in the context of event-based inputs and noisy sensing and computing.

The uni-directional projection between neurons in $L_{2/3}$ and $L_{5/6}$, on the other hand, preserves the learned temporal structure of the recurrent excitatory weights of $L_{2/3}$ and $L_{5/6}$. The temporal alignment observed in $L_{2/3}$ is present in $L_{5/6}$, even though we used the similarity in the uni-directional projection to sort the temporal activity of $L_{5/6}$ (see Figure 4.6).

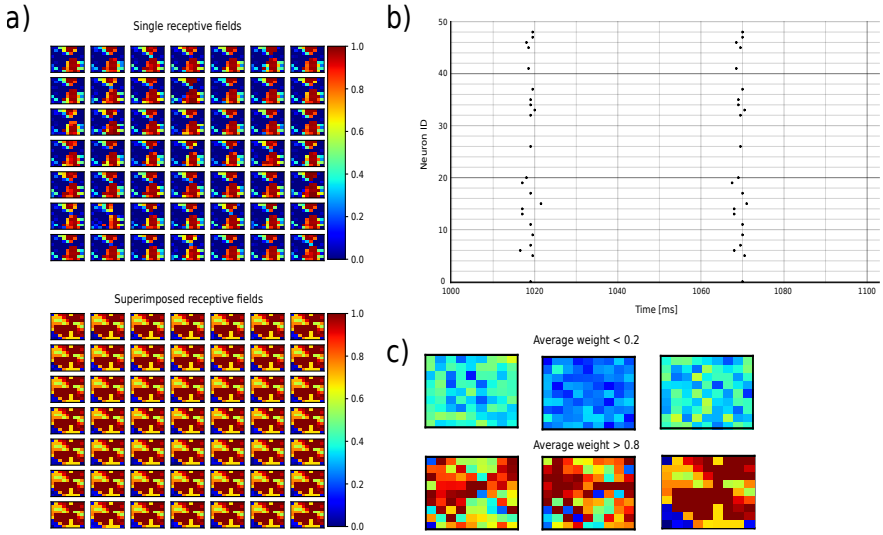


Figure 4.7: Example receptive fields and the effect of non-balanced network activity. a) Receptive field as learned by neurons in $L_{2/3}$ with σ_{target} set to zero (top) and set to one (bottom). A value of zero represents maximally strong inhibition is provided. This leads to the formation of a single receptive field which is shared among all neurons in $L_{2/3}$. A value of one indicates that all inhibitory synaptic weights will converge to zero and no inhibition is provided. The absence of inhibition leads to the formation of receptive fields which are characterised by the superposition of all prevalent orientations of the presented rotating bar. b) In the case of σ_{target} being one, only a single orientation is represented in the receptive fields which is also reflected in the spike raster plot. The activity is synchronised. Due to the too strong inhibition, the network starts to oscillate, which also explains the formation of a single receptive field. c) Example receptive fields which trigger the re-initialisation of the associated post-synaptic weight vector.

Stability and the effect of Activity Dependent Plasticity: The excitatory feed-forward and lateral recurrent projections enable the network to learn the spatio-temporal statistics of the presented input. The inhibitory projections are crucial in this context, as these synapses act as an activity regulariser while introducing competition between excitatory neurons. If we turned inhibition off, by setting σ_{target} to one, the learned receptive fields would represent a superposition of the input, thus being highly correlated (see Figure 4.7a, bottom). Furthermore, the network's activity would become temporally correlated and constant throughout the stimulus presentation. This is clearly due to STDP, as it is a correlation-based learning paradigm (Markram, Lübke, *et al.,* 1997; Bi & Poo, 1998; Song *et al.,* 2000). If no counter mechanism is present to decorrelate activity, STDP en-

forces already correlated activity even more. If we were, on the other hand, to set σ_{target} to zero, meaning that we install maximally strong inhibition similar to a connectivity probability of 1 between inhibitory and excitatory neurons, we would observe two effects. First, the network would start to oscillate, as the excitatory activity would lead to a massive injection of inhibitory currents back into the excitatory pool, which in turn would lead to strong inhibitory activity (see Figure 4.7b). The frequency of oscillation would be independent of the input and only determined by the synaptic time constants between the excitatory and inhibitory neurons (Van Vreeswijk *et al.,* 1994). Secondly, the oscillations thus introduced would lead to a single learned receptive field which is the same for all excitatory neurons, as the post-synaptic spike time, which is used by STDP, would be same for all excitatory neurons (see Figure 4.7a, top).

If we initialise the inhibitory weights with connectivity probability lower than 1, e.g. 0.7, and do not adjust the associated synaptic weight, the network would thereafter be capable of encapsulating the spatio-temporal statistics of the input and learning receptive fields for different orientation tunings (see Figure 4.3b & 4.4b). The static inhibition would lead, however, to oscillatory behaviour, as the excitatory weights increase due to STDP and introduce correlated activity which can not be compensated by the static inhibitory clamp (see Figure 4.4b). Furthermore, the static inhibitory clamp prevents the network from utilizing all its available computational units, e.g. neurons, because the associated inhibitory weight for a subset of excitatory neurons is too high (compare Figure 4.4b & c). If we use Equation 4.7 with intermediate, randomly distributed values of σ_{target} to regulate the strength of inhibitory synapses, we can effectively balance excitatory activity (see Figure 4.2), while allowing the network to use more neurons to tune their weights to learn sensible receptive fields and thus cover all presented orientations (compare Figure 4.4b & c and note the temporal overlap spikes of neuron 6 and 49). When combining ADP with heterogeneous network parameters, i.e. device mismatch, we effectively use more available computational resources, i.e. neurons, and prevent the network from starting to oscillate which would lead to a loss of information being encoded in the network (compare Figure 4.7b & 4.6).

If we would like to operate the network in rate-mode, we could rely on reverse STDP (Földiak, 1990; Roberts & Leen, 2010) to adjust the inhibitory weights. The inhibitory weight can only successfully be increased if the excitatory activity continuously proceeds the inhibitory activity since there is no direct causal link between inhibitory and excitatory activity, i.e. inhibition does not trigger

post-synaptic activity, but it instead prevents it. Neural activity, however, cannot be assumed to be constant over longer periods of time in the context of asynchronous sensing and processing.
Vogels and colleagues proposed an alternative learning paradigm to adjust inhibitory weights to balance ongoing processing in excitatory neurons (Vogels, Sprekeler, *et al.,* 2011). This inhibitory plasticity rule adjusts the inhibitory synaptic strength based on a pre-defined target rate. Their plasticity rule is similar to the ADP rule proposed here, however, their rate-based rule only uses post-synaptic super-threshold activity, i.e. spikes, to adjust the synaptic weights. The rule from Vogels does not consider sub-threshold membrane fluctuations, which are more likely to be present and informative in asynchronous inputs.
Both Vogels, Sprekeler, *et al.* (2011) and the learning rule proposed here adjusts synaptic weights to balance activity. Triesch (2005) proposed a mechanism, called intrinsic plasticity, which adjusts intrinsic parameters of the neuron model to ensure sparse activity, such that the overall activity stays bounded. However, the parameters which are adjusted change the transfer function of the neuron which transmits information solely in the output firing rate, but not in the precise timing of spikes. The adaptive thresholding approach of Afshar *et al.* (2019) seems more appealing in the context of event-based data to achieve the homoeostatic effects of intrinsic plasticity, as it was already demonstrated to work in the context of temporal-coded, time-continuous inputs.

Spike-Time Dependent Plasticity: The implemented Spike-Time Dependent Plasticity (STDP) rule was original proposed by Song *et al.* (2000). This plasticity rule captures weight update dynamics depending on pre- and post-synaptic spike times, i.e. pairs of spikes, found in biological neural networks (Markram, Lübke, *et al.,* 1997; Bi & Poo, 1998). Equation 4.5, however, only accounts for one specific temporal relation of pre- and post-synaptic spike pairs. In the last two decades, more temporal relations between pre- and post-synaptic spiking to update the synaptic weight have been experimentally measured (e.g. Abbott & Nelson, 2000; Roberts & Bell, 2002; Feldman, 2012; Graupner & Brunel, 2012). An interesting learning rule to account for more experimentally observed temporal weight update dynamics relies on triplets of spikes, i.e. post-pre-post instead of classical pairs of spikes (Pfister & Gerstner, 2006; Gjorgjieva *et al.,* 2011). The incorporation of triplet STDP represents an interesting addition to the system proposed here and will be subject to future research. Especially, it would be in-

teresting to compare the effect of classical and triplet STDP on the stability of the network and the incorporation of temporal sequences of learned spatio-temporal features.

4.4.1 Comparison to state-of-the-art event-based feature extraction

Events, as emitted by event-based sensors, can be converted into so-called time surfaces (see Section 1.4). These time surfaces can either be classified using echo-state networks (ESNs) (Lagorce, Ieng, *et al.,* 2015) or can be classified by template matching (Lagorce, Orchard, *et al.,* 2017; Afshar *et al.,* 2019). The feature templates can be stacked together to form a hierarchical feature extraction system (Lagorce, Orchard, *et al.,* 2017; Afshar *et al.,* 2019). The presence of a given time surface can either be conveyed (i) using integrate and fire neurons, which transmit information in their firing rate (Lagorce, Ieng, *et al.,* 2015) or can be conveyed (ii) in the form of an asynchronous AER events in which the payload encodes the presence of a given time surface (Lagorce, Orchard, *et al.,* 2017; Afshar *et al.,* 2019). Despite extracting spatio-temporal patterns from precisely timed events the output produced in the former case encodes information solely in rate-mode, meaning that for any downstream neuron the spike train does not carry additional information in its temporal domain. In the latter case, the time when a time surface event is generated carries information which can be used by downstream processing stage to classify moving (Afshar *et al.,* 2019) and stationary objects (Lagorce, Orchard, *et al.,* 2017), and even to perform face classification (Lagorce, Orchard, *et al.,* 2017).

Afshar *et al.* (2019) extended the template matching approach of Lagorce, Orchard, *et al.* (2017) by introducing adaptive thresholds, which provided faster convergence and more stable receptive fields. However, recurrent interactions between feature detectors serve only to introduce competition among the feature detectors (Lagorce, Ieng, *et al.,* 2015; Lagorce, Orchard, *et al.,* 2017; Afshar *et al.,* 2019) and need to happen in some cases instantaneously (Lagorce, Orchard, *et al.,* 2017; Afshar *et al.,* 2019). The instantaneous inhibition makes an implementation on spiking neuromorphic processors complicated and is biologically rather implausible.

We could demonstrate that excitatory recurrent synaptic projections when subject to learning carry information about the temporal sequence of presented stimuli, and that recurrent inhibitory projections can balance excitatory activity and thus drive the network into an efficient operation regime. The adaptive

threshold approach of Afshar *et al.* (2019) helped the network to use its computational resources and allocate them equally across available spatio-temporal features, similar to the proposed inhibitory plasticity mechanism. A combination of the adaptive threshold with plastic inhibition represents an interesting case study for future investigations.
The present work shows for the first time that ongoing processing can encapsulate an implicit representation of time, which emerges from the recurrent organisation of recurrently connected excitatory-inhibitory networks inspired by neocortical organisation principles. This temporal representation is preserved throughout processing and is thus also available for the next processing stage.

4.4.2 Limitations

The present study simulated only a single OCTA module and investigated the computational role of the feedforward, lateral recurrent and feedback projections within the OCTA module. When the interspike interval (ISI) of events presented inputs increases beyond the specified temporal window of the STDP mechanism, the feedforward and lateral recurrent connections, which encapsulate spatio-temporal features and their sequence, cannot be properly adjusted to incorporate the input statistics. This is just because no pair of spikes would fall into the temporal window of STDP and hence no potentiation or depression of any weight would take place. This scenario, however, is usually not the case for an event-based sensor, due to its time-continuous mode of operation and its high temporal resolution. Only if neither the environment nor the sensor moves, the ISI of events at the input will fall outside the temporal window of STDP in which case no learning should take place.

Spike-time dependent gain plasticity

We reported earlier that a single OCTA module learns phase shifted predictive fields with respect to the receptive fields. To understand the underlying cause for the observed phase shift between corresponding receptive and predictive fields of neurons in $L_{5/6}$, we systemically vary the angular velocity ω of the input and calculate the average phase shift $\Delta\Phi$ between corresponding preferred orientations of receptive and predictive fields (see Figure 4.8). The phase shift is learned via a plasticity rule which similar to classical STDP uses the temporal correlation of pre and post-synaptic spike timing, however, instead of learning a synaptic weight, the proposed plasticity modulates the synaptic gain (see

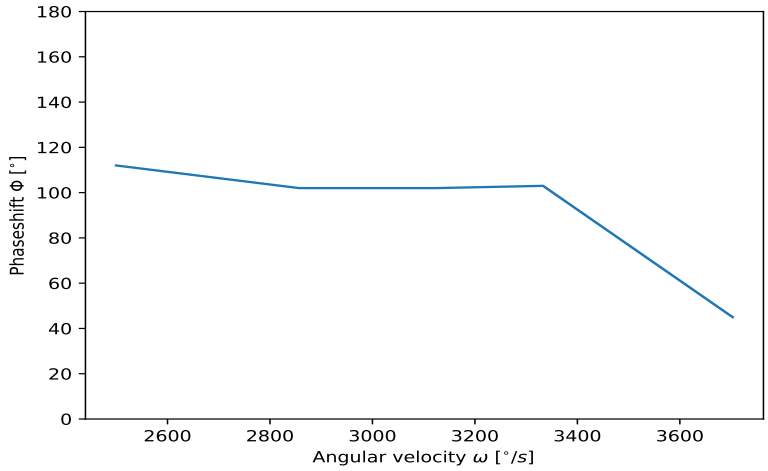


Figure 4.8: Phase shift between receptive and predictive fields of $L_{5/6}$ for different rotational velocities of the presented bar.

Eq. 4.8). The learned synaptic gain modulates the synaptic efficacy according to Equation 4.9. Hence, the information transmitted from L_4 to $L_{2/3}$ and $L_{5/6}$ respectively is gated, if the learned predictive field from $L_{5/6}$ temporally matches the input. When the rotational speed of the presented rotating bar is increased the represented phase shift between the learned, preferred orientations of corresponding receptive and predictive fields decreases. This phase shift is partially caused by the accumulated synaptic and neuronal delays of the system, thus the system has a preferred speed. This partially explains the constant phase shift angular velocities close to its preferred angular velocity in Figure 4.8. However, if the observed phase shift would be an effect of the accumulated system's delays only, the phase shift should be constant irrespective of the angular velocity. If synaptic weights become too weak, we re-initialise the synaptic weights (see Section 4.2.5). However, during this re-initialisation we keep the associated synaptic time constant fixed, meaning that at the moment the system can not learn optimal time constants for a given presented angular velocity (see Section 6.2.1 for possible mechanisms to learn synaptic time constants).
The synaptic gain modulation is based on precise spike timing, yet neuromorphic systems have certain limitations (see Sections 1.1 & 1.2 and Chapter 2). Hence, an algorithm emulated on neuromorphic hardware might be subject to temporal jitter, which in the context of the temporal predictions based on precise spike

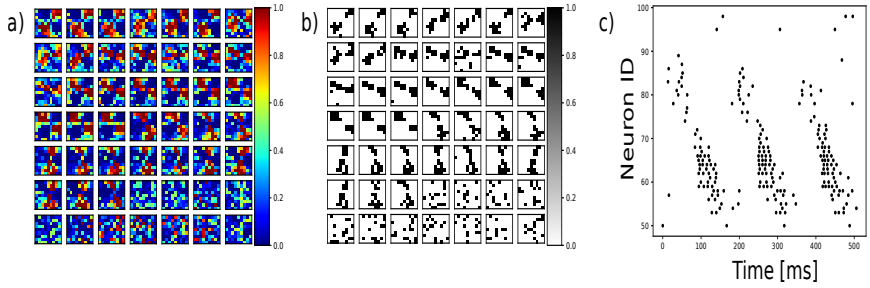


Figure 4.9: Receptive fields of $L_{2/3}$. a) Learned receptive fields of neurons in $L_{2/3}$ with 32-bit floating point precision. b) Quantised receptive field with 1-bit precision. Weights are quantised to meet limited weight resolution on the DYNAP. Each neuron only has 64 possible synaptic inputs. However, $L_{2/3}$ has in total three different weight matrices: feedforward inputs from L_4, recurrent inputs from $L_{2/3}$ and inhibitory inputs from the interneuron population. We quantise the weight matrix in such a way that only the 20 strongest weights are kept, whereas all others are pruned. c) Raster plot activity of $L_{2/3}$ emulated on the DYNAP. Raster plot is sorted according to the learned recurrent weights within $L_{2/3}$. Despite the one-step quantisation and a different mismatch distribution present on the DYNAP compared to the simulated mismatch the network shows the same temporal alignment as previously observed in software simulation. Note that the input is slowed down by a factor of 3 which has the effect that one revolution lasts 300 ms instead of 100 ms.

timing might be fatal. However, temporal jitter, i.e. uncertainty, may actually be beneficial, since no prediction, top-down attention, or bottom-up prediction, will be perfect in time and thus will never set the efficacy to the highest possible value (an avalanche of spikes), nor set the efficacy to the lowest possible value (killing the activity of the network completely).

4.4.3 Remarks for hardware emulation

A major motivation of the present work was to construct a neurally implemented algorithm to learn spatio-temporal features as emitted by a DVS, suited for neuromorphic processors. Analogue-digital sub-threshold neuromorphic systems, such as the DYNAP (Moradi *et al.,* 2018), have severe constraints on the bit-precision of synaptic weights (see Section 1.2). The simulations conducted in this study used, as is standard in computer simulations, a 32-bit floating point representation of synaptic weights. Secondly, adjacency matrices which were subject to plasticity were initialised with all-to-all connectivity. From a simulation point of view the all-to-all initialisation can easily be adjusted, when incorporating

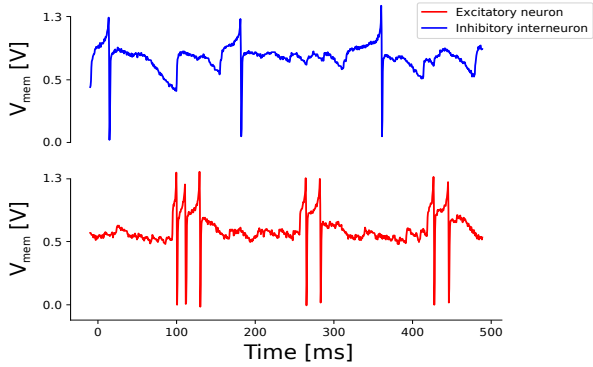


Figure 4.10: Example membrane potential traces of recurrently connected single excitatory and inhibitory neurons of $L_{2/3}$ emulated on the DYNAP. The learned inhibitory projections, whose weights were also quantised, are capable of balancing the excitatory activity. The recurrent excitatory inputs prime the neural activity (note excitatory ripples) and in conjunction with the feedforward activity elicit on average 1 to 3 spikes per neuron per preferred orientation. Note that the input is slowed down by a factor of 3 which has the effect that one revolution lasts 300 ms instead of 100 ms.

structural plasticity (e.g. George *et al.,* 2015; Spiess *et al.,* 2016). However, the mismatch between simulated weight resolution and the effective weight resolution of some neuromorphic processor is more severe (see Section 1.2).

Weight quantisation: To transfer parameters from a higher bit resolution to hardware which has a limited bit resolution compression scheme, such as quantisation, are used during training (Courbariaux *et al.,* 2015; Stromatias, Neil, *et al.,* 2015; Milde, Neil, *et al.,* 2017; Gysel *et al.,* 2018). A key feature of these quantisation schemes is that the available number of bits can be distributed dynamically for each parameter. In the context of neuromorphic processors, such as the DYNAP, we cannot afford to dynamically distribute the available number of bits due to weight sharing. The weight which is set by a bias current for each core on the DYNAP is shared among the 256 neurons present in this core (Moradi *et al.,* 2018). Furthermore, each synaptic input can only take 1 out of 4 absolute levels, i.e. $I_{WEIGHT_{EXC_F}}$, $I_{WEIGHT_{EXC_S}}$, $I_{WEIGHT_{INH_F}}$ or $I_{WEIGHT_{INH_S}}$ where I represents a bias current for the distinct excitatory (EXC), inhibitory (INH), fast (F) and slow (S) DPI synapse circuits (see Section 3.2.2 for schematics of theDPI synapse circuit and see Figure 9 in (Moradi *et al.,* 2018)). Furthermore, each

neuron on the DYNAP can only integrate 64 distinct synaptic inputs (Moradi *et al.,* 2018). Each synaptic projection within an OCTA module carries sensible information about the spatio-temporal statistics of the presented input. The total number of non-zero weights learned in the software simulation exceeds by far the 64 available synaptic inputs. Hence, we cannot use the the trick of emulating a higher weight resolution by varying the connectivity probability between two populations (Blum *et al.,* 2017; Milde, Blum, *et al.,* 2017; Milde, Dietmüller, *et al.,* 2017, & see 2).

Mapping learned weights on hardware: A single OCTA module has three synaptic projections (feedforward, recurrent excitatory and recurrent inhibitory) within $L_{2/3}$ and four synaptic projections (feedforward, recurrent excitatory, recurrent inhibitory and excitatory feedback) within $L_{5/6}$. To perform a first proof of concept of the applicability of parts of OCTA, i.e. $L_{2/3}$, on the DYNAP we follow a simple heuristic: Each synaptic projection can occupy 20 out of the 64 available inputs. We select 20 strongest weights by quantising the weights such that all other weights are set to zero (compare Figure 4.9a & b). We use the fast excitatory DPI circuit for the feedforward input synapses and the slow excitatory DPI circuit for the recurrent excitatory synapses. The inhibitory synapses use the fast inhibitory DPI circuit.

Emulating OCTA on hardware: After quantising all weight matrices projecting into $L_{2/3}$ we stimulate the DYNAP with the same stimulus and record the network's activity (see Figure 4.9c). The network activity shows the same temporal alignment along the different presented orientations. Note that we do not need to use a population average to cope with device mismatch. Furthermore, the inhibitory projections balance the excitatory activity, which is driven by the conjunction of recurrent and feedforward excitatory inputs (see Figure 4.10). This experiment confirms that simulating mismatch and optimizing the weights leads to not only stable network activity but also allows us to transfer learned synaptic weights to otherwise non-plastic neuromorphic processors. This finding will allow us in the future to explore different kinds of learning rules before we can build a system which incorporates the necessary learning mechanism directly in hardware.

4.4.4 Conclusion

We showed and argued how the proposed meta-architecture opens novel perspectives and understanding of how temporal information present in an event-based computing paradigm can be implicitly represented in ongoing processing.

Postamble

We were able to show, by incorporating event-driven plasticity mechanisms in recurrently connected networks of recurrently connected excitatory-inhibitory neuron populations, that this meta-network is capable of extracting statistically relevant spatio-temporal features and their temporal sequence from asynchronous event-based inputs. The proposed meta-architecture was inspired by the highly recurrent, stereotypically repeating canonical organisation structures found in the neocortex (see Section 4.2.1). Instead of making any prior connectivity assumptions within the network by hard-coding synaptic projections and hand-tuning their parameters, as we did in previous attempts to visual scene understanding (see Chapter 2 & 3), we installed unsupervised event-based plasticity rules such that the system can self-adjust its free parameters given the provided input (see Section 4.2.3).

The reason why we conducted this last study is two-fold: Firstly, we wanted to formulate a meta-architecture which is genuinely capable of learning spatio-temporal patterns irrespective of the sensory modality, but which only cares about precisely timed asynchronous streams of events. Secondly, we intended to construct a modular building block which can be (re-)used at different levels of feature complexity as its primary objective is to learn and predict spatio-temporal patterns. The installed prediction mechanism (see Section 4.2.4) can be understood as a locally encoded, self-generated error signal which in time-continuous systems is solely based on the precise time of predicted and incoming activity. This self-generated error signal can be seen as internally generated appropriate behaviour for predicting future inputs. The unsupervised nature of adjusting the network's parameters in combination with the recurrent organisation enabled the system to dynamically allocate its computational resources and installed an implicit temporal representation of time in ongoing processing.

We were able to demonstrate further that by incorporating inhomogeneous network parameters, i.e. device mismatch, in our simulation the system not only successfully prevents fatal oscillatory states due to correlation-based learning rules (see Figure 4.6 & 4.5), but more importantly that the learned parameters, i.e. synaptic weights, can be successfully mapped onto mixed-signal sub-threshold neuromorphic processors (see Figure 4.9 & 4.10).

Thereby, we achieved a formulation of a feature extraction system suited for event-based neuromorphic sensory-processing, to potentially learn spatio-temporal features relevant for object recognition and forming high-level semantic

concepts. Furthermore, we could show for the first time that device mismatch is a crucial feature for stable learning and computation and when combined with event-based plasticity rules can be exploited to allocate computational resources whose parameters best match the provided inputs.

PART IV

The octopus in the real world

5

Discussion

VISUAL scene understanding is, among others, one crucial requirement for a system to successfully navigate or identify objects of interest. In this thesis we started our journey towards visual scene understanding by discussing how biological systems need to continuously perceive changes at the sensory level to construct a stable visual perception, and that spiking neurons meet the requirements to operate on such time-continuous changes even if these changes are sensed asynchronously. We discussed further that networks of spiking neurons have two ways of representing signals: information inside the network is either conveyed within the output rate of any given neuron, i.e. with a *rate code*, or within the time of any elicited spike or the relative timing between pairs of spikes, i.e. with a *temporal code* (see Chapter 1). Our current understanding of computation in Artificial Neural Networks (ANNs), however, is primarily a rate-code understanding. Although mixed-signal neuromorphic sensory-processing systems (see Section 1.1 & 1.2) can be operated in a purely rate-coded fashion, these systems can be operated also in a temporally-coded fashion, due to their communication protocol and their asynchronous and fully distributed nature. To achieve a better understanding of the computational benefits of temporal coding for the representation of information, we used mixed-signal neuromorphic sensory-processing systems, since, in these systems, time represents itself. Thus, we developed in this thesis neurally implemented algorithms which, besides rate-coding, also convey information in the temporal dimension of their processing, not only to leverage the full potential of Spiking Neural Networks (SNNs) emulated on neuromorphic processors, but also to gain a better understanding of the temporal aspect of neural computation.

On the quest to enable neuromorphic sensory-processing systems to visually un-

derstand scenes we first showed that these systems are capable of the simplest form of visual scene understanding: Generating appropriate behaviour given visual inputs such that the system does not collide with objects in unknown environments. To improve the system's capability of understanding visual scenes and to study how temporal-coding can be beneficial to do so, we equip the system in a second step with a sense of motion and distance. In a last step, to enable neuromorphic sensory-processing to genuinely understand visual scenes, we allow the system to adjust its parameters based solely on the spatio-temporal statistics of its input. Thus we equip the system with an implicit temporal representation of time in ongoing processing and demonstrate that the previously identified limitations of neuromorphic sensory-processing systems are crucial features for stable computation in SNNs. For an in-depth discussion and comparison to the state of the art please refer to Sections 2.4, 3.5 & 4.4, respectively.

Reactive control: The system presented in Chapter 2 represents a neurally implemented Braitenberg-vehicle in which the decision tree of how to avoid collisions with objects is hard-coded (see Figure 2.3 & 2.15). The spatial origin of incoming events primarily drives the behaviour of the system. We could show that the inhomogeneity due to mismatch and thermal noise present in neuromorphic sensory-processing systems could be compensated in this system by using populations of neurons sharing the same input and targeting the same outputs. By encoding information in small populations rather than single neurons, we could further demonstrate that the limited precision of synaptic weights in neuromorphic processing systems (see Section 1.2 & 2.2) can be compensated by installing multiple synaptic connections between neuronal populations if a stronger effective weight is needed. This not only increased the effective weight precision, but it also guaranteed smooth avoidance behaviour as the peripheral pixels had fewer synaptic connections, and thus weaker effective weights, compared to central pixels (see Figure 2.15a). Both of the aforementioned architectural improvements enabled the system to interact with its environment in real-time, despite noisy sensory inputs and noisy computing units. Although the system represents an important stepping stone towards robotic controllers and visual scene understanding, realised directly in analogue neuromorphic hardware, it does not address how to incorporate temporal information implicitly in ongoing processing and computation. The reason for this is due to its neural architecture, which solely uses the spatial origin of detected events to ad-

just avoidance manoeuvres. The system does not explicitly extract information present in the temporal domain of its input, meaning that the neural architecture does not consider the point in time when an event is emitted as informative, nor does it encode any behavioural relevant information in the timing of elicited spikes. Information transmitted within the system and to control the motors is encoded in the output firing rates.

A sense of motion and distance: Although the previous study represents a proof of concept for closed-loop real-time visual scene understanding, a key feature of mixed-signal sub-threshold neuromorphic sensory-processing systems is their ability to represent precise temporal information naturally. A sensible computation in the temporal domain in the context of navigation is estimating temporal correlations. When an object moves across the retina and an algorithm estimates the temporal correlation of spatially adjacent inputs, the estimated correlation is proportional to the amplitude of optic flow (see Section 3.1.1). Translational optic flow scales inversely with the relative distance to an object, thus the distance estimate can be used in the context of navigation to scale avoidance manoeuvres based on relative spatio-temporal information rather than pure spatial information. We designed a small and low-power temporal correlation detector, called the spiking Elementary Motion Detector (sEMD), using a single neuron and synapse whose efficacy was modulated with an exponentially decaying time-varying signal triggered by the first and facilitated by the second spike of a pair spikes (see Section 3.2.2). The sEMD represents the first motion detector which preserves the temporal correlation information present at its input, in its output (see Figure 3.7 and Section 3.5). The sEMD additionally encodes the amplitude of optic flow in its output rate (see Figure 3.5) and is suited for neuromorphic processing systems. A downstream network of spiking neurons can directly use the output of the sEMD. We showed that the underlying principle could be both simulated using CPUs or digital neuromorphic processors such as SpiNNaker, and emulated in analogue/digital sub-threshold neuromorphic circuits. We further demonstrated that the sEMD can be used to successfully navigate in open- as well as closed-loop conditions (see Figure 3.8 & 3.10).
Computing temporal correlations not only equip a system with a better understanding of its visual environment as it has access to information about the 3D structure but more interestingly the underlying principle of the sEMD represents a neural algorithm which extracts information purely from relative temporal

information. It thus can be seen as a first step towards temporal computation in neural networks, as information is extracted from temporally coded inputs and the output transmits information in rate and temporal code.

Online Clustering of Temporal Activity: We proposed in Chapter 3 the sEMD model at whose core a Adaptive Synaptic Efficacy (ASE) mechanism modulated the synaptic transmission efficacy based on the precise relative timing of an incoming pair of spikes. We hypothesised that the proposed mechanism for temporal correlation detection could be used to route information in SNNs dynamically. Although the sEMD represents a first neural algorithm which is capable of temporal computation suited for event-based neuromorphic sensory-processing systems, it was embedded in a hard-coded network architecture. To overcome this limitation and thus build a system capable of genuinely visual scene understanding we built a system inspired by the stereotypically repeating, highly recurrent canonical microcircuits of the neocortex (see Figure 4.1). We incorporated the ASE mechanism in our meta-architecture, in which, unlike the up-regulation of EPSCs of Chapter 3, it down-regulated synaptic transmission efficacy in the presence of temporally correlated pairs of spikes (see Figure 4.2). The objective of the proposed meta-architecture is to learn spatio-temporally correlated patterns of incoming events, while also learning their temporal sequence, thus incorporating an implicit temporal representation of time in ongoing processing (see Figure 4.6 & Section 4.4). In other words, the proposed meta-architecture performs Online Clustering of Temporal Activity, which we thus named (OCTA) This intrinsic representation of time enabled an OCTA module to perform temporal predictions of incoming events using the ASE mechanism. A single OCTA module represents a general building block, which can be understood as temporal autoencoder implemented with spiking neurons and suited for event-based inputs. Furthermore, due to incorporating inhomogeneous network parameters we were not only able to show that device mismatch is beneficial for stable computation in SNNs, but more importantly it allowed us to verify the applicability of parts of a single OCTA module on the current generation of mixed-signal sub-threshold neuromorphic processors, such as the DYNAP (Moradi *et al.,* 2018). Since we already showed the working behaviour of the ASE mechanism in mixed-signal sub-threshold circuits (Milde, Bertrand, Ramachandran, *et al.,* 2018, & see Section 3.2.2), we can potentially in the future incorporate an entire OCTA module in the next generation of mixed-signal sub-threshold neuromorphic processors.

This would allow us to study learning paradigms and additional stability mechanism, such as adaptive thresholds (Afshar *et al.,* 2019), and verify the effect of these additions in hardware. Furthermore, we can study the computational advantages of implicit temporal representation of time in SNNs and how we can use this representation as a self-generated teacher signal directly in hardware. The last study of this thesis represents a starting point and a first proof of concept towards computation in SNNs that takes advantage of the inherently available temporal dimension and information transmission in SNNs based on temporal-coding.

6

Scientific contribution

From the presented neural algorithms in this thesis, two core mechanisms emerged which are unique and open novel perspectives on neural computation in Spiking Neural Networks each reflecting some aspect of computation in the temporal mode of operation: The Adaptive Synaptic Efficacy and the online clustering and predictive learning mechanism. These mechanisms are important stepping stones towards enabling SNNs to visually understand scenes and take advantage of their additional dimension compared to ANNs: Time.

6.1 Adaptive Synaptic Efficacy mechanism

By adaptively scaling synaptic efficacy, i.e. synaptic gain, based on a time-varying signal, a single conventional DPI synapse acts as a temporal correlation detector which can be used to estimate optic flow, estimate inter-aural time differences and dynamically gate information in SNNs. The proposed gating mechanism opens a novel theory of attentive and predictive neural computation which is solely based on a temporal code and demonstrates how SNNs can alternatively convey information in their ongoing processing. We were able to show that the underlying ASE mechanism can be used for up- and down-regulating synaptic transmission efficacy and how the core principle can install attention- and prediction-signals based on the precise relative timing of spikes. If this mechanism would be incorporated in larger-scale networks, amplification, i.e. up-regulating transmission efficacy would shift the network's attention on the basis of activity towards activity which matches desired expected activity and the actual input. On the other hand, attenuation, i.e. down-regulating transmission efficacy would lower the overall network's activity, as the internal network dy-

namics can account for the observed incoming activity as long as the predictive signals match the actual inputs. This way of dynamically gating information in SNNs would lower the overall number of generated spikes leading to spatio-temporal sparse activity in the network and thus lower the overall power consumption of the system Moreover, it would naturally, in the context of online learning and processing, lead to parameter updates as long as attentive or predictive signals do not match the actual inputs. In this context, any source of temporal uncertainty, i.e. temporal jitter, would be in fact beneficial for the system as this uncertainty would prevent the system from triggering fatal, large-scale avalanches of activity or cancelling all activity at once given a perfect match of attention or prediction signals, respectively.

6.2 Online clustering and predictive learning mechanism

By providing recurrently connected populations of excitatory and inhibitory neurons with the ability to adjust their synaptic parameters, these networks are able to capture statistically relevant spatio-temporal information and cluster a high dimensional input space into a lower dimensional feature space. The underlying learning mechanisms are solely based on locally available information and completely event-driven, thus suited for event-based neuromorphic sensory-processing systems. As the system not only learns feedforward receptive fields but also adjusts its recurrent connections to capture temporal sequences and use these temporal representations to learn feedback predictive fields, the system encodes local, self-generated error signals, which enables the system to be operated continuously, i.e. online. We were able to show for the first time that when network parameters are adjusted according to the presented inputs, device mismatch, present in mixed-signal sub-threshold neuromorphic systems, is a feature that can stabilise the network's activity, which does not need to be compensated by a population average. Device mismatch, which is manifested as inhomogeneous parameters sampled from a Gaussian distribution, successfully prevents the network from oscillatory firing which otherwise leads to synchronous network activity and thus to homogeneous receptive and predictive fields where the timing of generated spikes does not carry any information anymore. The proposed inhibitory plasticity rule is not only suited for SNNs operated using a *temporal code*, but more interestingly introduces spatio-temporal competition and excitatory-inhibitory balance, while allowing the network to use more computational resources and to attach meaning to the time of elicited spikes. This

inhibitory plasticity mechanism can be understood as indirectly adapting the neuronal firing threshold, but the homoeostatic effect, on the other hand, emerges from internal network dynamics. The proposed plasticity rule is, in contrast to adapting neuronal firing thresholds, solely based on locally available information and does not need any semi-global information from adjacent neurons or synapses.

The last study of this thesis shows that the timing of spikes does carry information if the system is operated in a time-continuous manner and subject to event-based inputs, in which the time of occurrence of an event is the primary descriptor, besides its inherently encoded payload. The precise timing of the input can be used to learn spatio-temporally correlated patterns which manifest themselves in precisely timed "output-spikes", thus enabling the network to convey temporal information between neuronal populations. This study, furthermore, showed how the precise timing of feedback spikes could be used as a self-generated teacher signal, orchestrating information transmission in Spiking Neural Networks. The recurrent organisation principles used to study the computational role of time in ongoing processing allowed the network to capture longer temporal dynamics than could be explained by the longest synaptic time constant. These findings provide novel perspectives on hierarchical neural computation, as lateral and top-down synaptic projections, in addition to providing competition and signal restoration, carry temporal, task-relevant information on much longer time scales.

6.2.1 Online Clustering of Temporal Activity Predictive Understanding of Scenes

A necessary next step towards hierarchical, time-based neural computation and visual scene understanding is to study the computational implications of top-down modulation and the capabilities of the proposed gating mechanism in the presence of laterally moving stimuli across multiple OCTA modules. Although this is beyond the scope of this thesis, we would like to lay out a roadmap towards a system operating on multiple distinct visual input receptive fields and hierarchically organised OCTA modules. This extension would lead to Online Clustering of Temporal Activity for Predictive Understanding of Scenes (OCTAPUS) framework. A crucial addition to the already incorporated plasticity mechanisms for synaptic weights would be to adjust synaptic time constants based on e.g. structural plasticity (Bailey & Kandel, 1993; Gerstner *et al.,* 1996; Holtmaat &

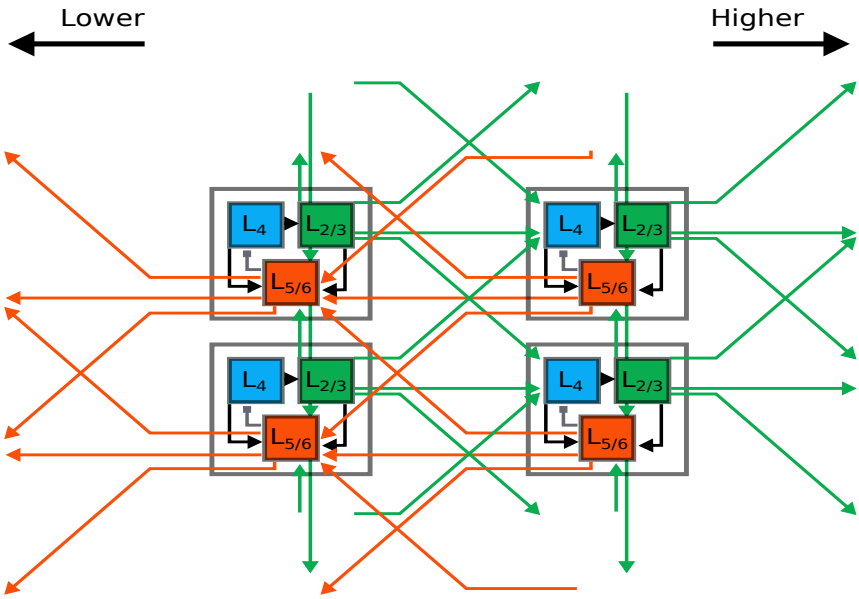


Figure 6.1: Inter-module connectivity scheme between multiple OCTA modules organised in a hierarchy. Green arrows indicate feedforward and lateral recurrent projections, whereas orange arrows represent top-down feedback projections. Note that inhibitory interneuron populations are not depicted for better visibility.

Svoboda, 2009; George *et al.,* 2015; Spiess *et al.,* 2016). Furthermore, by incorporating activity-dependent plasticity rules to adjust neuronal time constants, the system would have access to a richer repertoire of network dynamics to better capture provided inputs (Soldado-Magraner *et al.,* 2017). Both additions would enable the system to faithfully learn phase or spatio-temporally shifted predictive fields on longer time scales and broader rotational and translational velocity ranges. To study the effect of translational moving stimuli across multiple adjacent receptive fields and the effect of top-down predictions and activity modulation we need to assemble multiple OCTA modules in a hierarchically distributed network of modules (see Figure 6.1). This will allow us to provide real-world stimuli to the system and study the computational implications of lateral and top-down predictions. Efferent projections of $L_{2/3}$ provide inputs to adjacent $L_{5/6}$ and to upstream L_4 populations, whereas efferent projections of $L_{5/6}$ target local L_4 and down-stream $L_{5/6}$ populations. We were able to show that a given synaptic projection, e.g. $L_{2/3}$ to $L_{5/6}$, is able to preserve the temporal structure, which is encoded in the respective lateral recurrent weight matrix.

Although this represents only a first proof of concept that learned temporal statistics could be conveyed between different populations of neurons, it should be possible to do the same across different hierarchical levels. An interesting addition to the proposed system would be the incorporation of the model equivalent of superficial cortical layer L_1 to provide long-range, top-down attentive inputs. In this context, it is important to start thinking about events transmitted within the system as messages, which inherently carry a certain payload, e.g. spatial origin or the presence of a certain spatio-temporal feature, and construct appropriate payload-independent and event-based learning rules.

7

Conclusion

On our quest to enable neuromorphic sensory-processing systems to visually understand their environment we developed three neural algorithms to extract task-relevant cues to generate appropriate collision avoidance behaviour and to predict future visual inputs. The first system we developed is a low-level reactive control architecture which successfully compensated device mismatch and increased the effective weight resolution by varying the connectivity probability between neuronal populations. The second system extended this reactive approach to scene understanding by incorporating a neural algorithm which extracts relative distance information solely based on the relative timing of pairs of spikes by calculating the temporal correlation of these spikes. The third system we developed is capable to learn and predict spatio-temporal patterns by adjusting free network parameters in a completely unsupervised, event-driven manner, based solely on locally available signals.
These systems represent important stepping stones towards a better understanding of the computational advantages when we consider the timing of spikes as being informative.

Benefits of hardware emulation: The verification of the here presented neural algorithms by emulating them on neuromorphic hardware served two major purposes: (i) to provide a proof of principle and (ii) to study the algorithmic benefits of the present hardware limitations. The former purpose helped us to make sure that the proposed algorithms are in fact suited for their application to the event-driven sensory-processing paradigm. The latter purpose not only allowed us to save resources such as synaptic connections, but also enabled the simulated and emulated recurrent neural networks to perform stable computation due to

the introduction of device mismatch which is reflected in heterogeneous model parameters and transfer functions.
The limited number of incoming synaptic projections present on the DYNAP had the consequence of performing quantisation of synaptic weights. By quantising the synaptic weights we re-normalised the synaptic weights indirectly which had the consequence that the receptive fields showed a much cleaner qualitatively orientation preference (compare Figure 4.9a & b). The present device mismatch on the other hand naturally led to a desynchronisation of the network's activity and thus prevented task-irrelevant oscillations from dominating the network's activity and hence the incorporated information. The main advantage of emulating neural networks on mixed-signal analog/digital neuromorphic hardware, however, is that it enabled us to understand the effect, impact and contributions for computations of certain synaptic projections, as a change in the bias current which determines the synaptic weight can be observed in real-time. This feature of the specific neuromorphic hardware platform used allowed us to vary the synaptic strength of the recurrent excitatory synaptic projections and let us observe that only in conjunction with the excitatory feedforward synapses neurons in layer $L_{2/3}$ showed the temporal alignment of neural activity which can be seen in Figure 4.9. Meaning that the recurrent synaptic projections not only incorporate a temporal sequence, but also prime other excitatory neurons and perform temporal pattern completion.

Possible experimental predictions: In Chapter 4 we demonstrated by employing pre-existing and developing novel event-based learning rules which update respective synaptic parameters, such as the synaptic weight, based on relative spike timing that the network's activity was sparse along the stimulus presentation. Meaning that for a particular presented orientation less than a hand full of neurons were active, thus covering a full stimulus revolution smoothly and spikes are almost evenly distributed in time. This temporal sparsity represents a very interesting emergent property which results from the adaptation of especially inhibitory synaptic weights. It not only implies that the overall power consumption is lowered as the number of spikes is lowered, but also it enables the system to attach meaning to particular spikes as the available time is used by the system to communicate the presence of a detected orientated edge and their sequences. The temporal sparsity is a natural consequence if the recurrent neural network extracts its incorporated information from precise spike timing. When we assume that temporal precision not only is a consequence of sensing

the environment with sub-millisecond temporal precision, but temporal precision is also a consequence of actively generated movements an experimental prediction would be that the temporal variability of spikes to a spatio-temporal pattern should be smaller with a higher degree of temporal sparsity in awake behaving animals compared to anaesthetised animals with simulated saccadic eye movements ($< 15ms$, for anaesthetised experiments please refer to Baudot *et al.,* 2013). Moreover, when incorporating micro-saccadic eye movements the temporal variability of spike times should be lowered even more.
Leinweber *et al.* (2017) showed that mice can re-learn visuo-motor relations in a controlled virtual reality set-up. Another experimental prediction one could make based on the results presented in Chapter 4 is that the time it would take for a mice to re-learn spatio-temporal visuo-motor relationship should be longer if the animal is anaesthetised and thus only passively observing its environment compared to an awake behaving animal as reported in Leinweber *et al.* (2017).

On the potential and limitation of processing time: In an ideal system the length of a learned sequence would purely be determined by the number of neurons present in a given population, as each neuron would represent one state within the sequence and the recurrent weights represent the transition probability between these states (Kappel *et al.,* 2014). However, this is certainly a theoretical optimum which avoids redundancy and assumes very high bit-precision of synaptic weights, e.g. 64-bits. In biological neural networks, as well as in neuromorphic processing systems, however, redundancy is a crucial requirement as it ensures computation even in the presence of cell death or circuitry failure. Additionally, redundancy enables a network, whether biological or artificial, to deal with heterogeneity in their processing elements and with limited precision of synaptic weight which is thought to be lower than 5 bits in biological synapses (Bartol Jr *et al.,* 2015). Furthermore, if one would initialise synaptic projections rather sparse the absolute number of synaptic connections would be limited which would again constraint the maximally possible sequence length.
In Chapter 4 we could learn a sequence of 36 items. What the practical limitation is, is subject to future research. A potential stimulus to test the maximal length of a sequence would be to decrease the angular step size of the rotating bar but keeping the ISI of presented orientations fixed in combination with systematically varying the number of available neurons in for example layer $L_{2/3}$.
As a single OCTA module already incorporated a sequence of events across spatially distributed channels, stacking OCTA modules in a hierarchy as depicted in

Figure 6.1 should naturally lead to learning sequence of sequences. This hierarchical stacking would also allow a network of OCTA modules to incorporate time scales which exceeds the present synaptic and/or neuronal time constants present in any given OCTA module.
When the stimulus is moving too fast compared to the synaptic, neuronal and/or STDP time constants on the other hand, a single OCTA module will fail to pick up correlated features. The only possible solution for this failure case would be re-tuning the respective time constants to meet the speed of the fasted presented spatio-temporal sequence.
In contrast, when the mean of synaptic and/or neuronal time constant reflect the temporal statistics coarsely a possible exploration to learn matched synaptic time constants is to not only re-initilise synaptic weights when these weights drop below some user specified threshold as described in Section 4.2.5, but also to re-sample the associated synaptic time constant. While this synaptic time constant sampling has already been propesed by Gerstner *et al.* (1996) and can be functionally understood as structural plasticity (Bailey & Kandel, 1993; Holtmaat & Svoboda, 2009; George *et al.,* 2015; Spiess *et al.,* 2016), adjusting the neuronal time constant could also greatly benefit the present model to incorporate a multitude of time scales, (e.g. Triesch, 2005; Soldado-Magraner *et al.,* 2017). The formulation of event-driven learning rules to adjust neuronal time constants based on relative timing information is subject to an ongoing collaboration and research.

Future work: Possible next steps for future investigations could include the following:

1. Single translating bars within a single receptive field
2. Two orthogonal translating bars with a single receptive fields (at different speeds)
3. Translating bars across multiple receptive fields, to study lateral interactions of multiple OCTA modules
4. Design a cyclic stimulus which undergoes a bifurcations with a certain probability and investigate if the switching probability is also included in recurrent synaptic weights
5. Present more spatial complex satio-temporal patterns to emphasise the need for a hierarchical feature extraction network

The list above provides a possible set of stimuli which could be used to study the limitations and possibility of a single or multiple OCTA modules. In combination with additional learning rules mentioned in the paragraph above the present synaptic and/or neuronal learning mechanisms can be tested and verified.

A possible application of the proposed OCTA architecture could be to learn spatio-temporal features in order to track satellites in the context of space situational awareness problems or identify underwater mines in the context of deployable submersible drones. In both task context it would be extremely interesting to investigate a combination of existing event-based feature extraction algorithms like those developed by Lagorce, Ieng, *et al.* (2015) or Afshar *et al.* (2019) with the recurrent plasticity introduced within an OCTA module, as well as across OCTA modules.

Impact of this work: The development of the spiking Elementary Motion Detector (sEMD) in Chapter 3 not only led to the design of a test chip, but more importantly represents the theoretical foundation which served as a starting point for a master's thesis with the objective to develop a real-time, closed-loop robotic collision avoidance system based on optic flow extracted on a neuromorphic processor. Secondly, the development of the software infrastructure, called teili, used to simulate the recurrent SNN presented in Chapter 4 was primarily started by Moritz Milde and Alpha Renner. Currently, eight people are actively contributing to teili by developing new tools, functionalities and backends, while more than 15 people are using teili on a daily basis to test and simulate different neural algorithms as close to the hardware as possible. Teili provides already functions to simulate device mismatch and to semi-automatically emulate previously simulated SNNs on neuromorphic hardware, such as the DYNAP (Moradi *et al.,* 2018).
The developed software infrastructure will ease the development of novel neural algorithms and also represents a playground for junior and senior scientists to explore different combination of neural algorithms in a more compact and easier to use fashion. While the last two points serve primarily scientific research, the developed software infrastructure will enable the community to more easily port neural algorithms to a variety of neuromorphic processors, but also to apply previously developed neural algorithms to other sensory modalities in a more reproducible manner.

Summary and implications: We could demonstrate that a single synapse can preserve temporal information present at its input, in its output. We could further demonstrate that recurrent Spiking Neural Networks (SNNs) are capable of incorporating an implicit temporal representation of time in their ongoing processing. The implicit representation of time in the information transmission

within a SNN opens a new perspective on attention and how information can be effectively amplified or attenuated within the network. By utilising recurrent excitatory and inhibitory synaptic projections we achieved (i) to balance activity and (ii) embed temporal expectations. This enabled a recurrently connected network of excitatory-inhibitory neurons to represent some notion of time. The time scale at which temporal information is being represented exceeds what could be explained by the synaptic and neuronal time constants only. This opens a new perspective on learning and memory formation across different time scales primarily due to recurrent organisation principles. Thus, I hope I could shine a bit more light onto the maybe greatest remaining mystery: The nature of time.

Bibliography

1. Cauchy, A. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris* **25,** 536–538 (1847).
2. Marey, E.-J. *Physiologie du mouvement: vol des oiseaux* (G. Masson, 1890).
3. Adrian, E. *The Basis of Sensation: The Actions of the Sense Organs* (ed Christophers, L.) (Lowe and Brydone Printers LTD., London, 1928).
4. Gibson, J. *The Perception of the visual world* (Boston, MA: Houghton Mifflin, 1950).
5. Robbins, H. & Monro, S. A Stochastic Approximation Method, Annals Math. *Statistics* **22,** 400–407 (1951).
6. Hodgkin, A. & Huxley, A. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology* **117,** 500–44 (1952).
7. Hassentstein, B. & Reichardt, W. Systemtheoretische analyze der Zeit-Reihenfolgen- und Vorzei-chenauswertung bei der Bewegungsperzeption des Rüsselkäfers chlorophanus. *Z. Naturforsch.* **11b,** 513–524 (1956).
8. Von Neumann, J. *Probabilistic logics and synthesis of reliable organisms from unreliable components* in *Automata Studies* (eds Shannon, C. & McCarthy, J.) (Princeton University Press, 1956), 43–98.
9. Coombs, J., Curtis, D. & Eccles, J. The generation of impulses in motoneurones. *The Journal of physiology* **139,** 232–249 (1957).
10. Bryson, A. E. & Denham, W. F. A steepest-ascent method for solving optimum programming problems. *Journal of Applied Mechanics* **29,** 247–257 (1962).

11. Hubel, D. & Wiesel, T. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Jour. Physiol.* **160,** 106–54 (1962).

12. Barlow, H. & Levick, W. The mechanism of directionally selective units in rabbit's retina. *The Journal of physiology* **178,** 477–504 (1965).

13. Hubel, D. & Wiesel, T. Receptive Fields and Functional Architecture in the Two Nonstriate Visual Areas (18 and 19) of the Cat. *Jour. Neurophysiol.* **160,** 106–154 (1965).

14. Chua, L. Memristor-the missing circuit element. *Circuit Theory, IEEE Transactions on* **18,** 507–519 (1971).

15. Konishi, M. Sound Localization in the Barn Owl. *The Journal of the Acoustical Society of America* **50,** 148–148 (1971).

16. Thomas Jr, C., Springer, P., Loeb, G., Berwald-Netter, Y. & Okun, L. A miniature microelectrode array to monitor the bioelectric activity of cultured cells. *Experimental cell research* **74,** 61–66 (1972).

17. Wilson, H. & Cowan, J. A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Biological Cybernetics* **13,** 55–80 (1973).

18. Chua, L. & Kang, S. Memristive devices and systems. *Proceedings of the IEEE* **64,** 209–223 (1976).

19. Stevens, J. K. *et al.* Paralysis of the awake human: visual perceptions. *Vision research* **16,** 93–IN9 (1976).

20. Hubel, D. & Wiesel, T. *The functional architecture of the macaque visual cortex. The Ferrier Lecture* in *Proc. R. Soc. Lond. B* **198** (1977), 1–59.

21. Backus, J. Can programming be liberated from the von Neumann style?: a functional style and its algebra of programs. *Communications of the ACM* **21,** 613–641 (1978).

22. Essen, D. v. & Zeki, S. The topographic organization of rhesus monkey prestriate cortex. *The Journal of physiology* **277,** 193–226 (1978).

23. Gibson, J. *The ecological approach to visual perception* (Boston, MA: Houghton Mifflin, 1979).

24. Dean, A. The variability of discharge of simple cells in the cat striate cortex. *Experimental Brain Research* **44,** 437–440 (1981).

25. Hamill, O. P., Marty, A., Neher, E., Sakmann, B. & Sigworth, F. Improved patch-clamp techniques for high-resolution current recording from cells and cell-free membrane patches. *Pflügers Archiv* **391,** 85–100 (1981).

26. Lucas, B. & Kanade, T. *An Iterative Image Registration Technique with an Application to Stereo Vision* in *IJCAI81* (1981), 674–679. citeseer.ist.psu.edu/lucas81iterative.html.

27. Lelkens, A. & Koenderink, J. J. Illusory motion in visual displays. *Vision Research* **24,** 1083–1090 (1984).

28. Foster, K., Gaska, J. P., Nagler, M. & Pollen, D. Spatial and temporal frequency selectivity of neurones in visual cortical areas V1 and V2 of the macaque monkey. *The Journal of physiology* **365,** 331–363 (1985).

29. Braitenberg, V. *Vehicles: Experiments in Synthetic Psychology* ISBN: 0262521121. doi:10.2307/2185146. arXiv: arXiv:1011.1669v3 (MIT press, 1986).

30. Mysliwetz, B. D. & Dickmanns, E. D. *A Vision System with Acitve Gaze Control for Real-Time Interpretation of Well Structured Dynamic Scenes* in *Intelligent Autonomous Systems, An International Conference* (1986), 477–483.

31. Rumelhart, D. E., Hinton, G. E., McClelland, J. L., *et al.* A general framework for parallel distributed processing. *Parallel distributed processing* **1** (1986).

32. Lund, J. S. Local circuit neurons of macaque monkey striate cortex: I. Neurons of laminae 4C and 5A. *Journal of Comparative Neurology* **257,** 60–92 (1987).

33. Douglas, R., Martin, K. & Whitteridge, D. A Canonical Microcircuit for Neocortex. *Neural Computation* **1,** 480–488 (1989).

34. Mead, C. *Analog VLSI and Neural Systems* (Addison-Wesley, Reading, MA, 1989).

35. Pelgrom, M., Duinmaijer, A. & Welbers, A. Matching properties of MOS transistors. *IEEE Journal of Solid-State Circuits* **24,** 1433–1440 (Oct. 1989).

36. Földiak, P. Forming sparse representations by local anti-Hebbian learning. *Biological cybernetics* **64,** 165–170 (1990).

37. Ogawa, S., Lee, T.-M., Nayak, A. S. & Glynn, P. Oxygenation-sensitive contrast in magnetic resonance image of rodent brain at high magnetic fields. *Magnetic resonance in medicine* **14,** 68–78 (1990).

38. Thorpe, S. J. Spike arrival times: A highly efficient coding scheme for neural networks. *Parallel processing in neural systems,* 91–94 (1990).

39. Brooks, R. A. New approaches to robotics. *Science (New York, N.Y.)* **253,** 1227–1232. ISSN: 0036-8075 (1991).

40. Horiuchi, T., Lazzaro, J., Moore, A. & Koch, C. *A delay-line based motion detection chip* in *Advances in neural information processing systems* **3** (1991), 406–412.

41. Douglas, R. & Martin, K. A Functional Microcircuit for Cat Visual Cortex. *Jour. Physiol.* **440,** 735–769 (1992).

42. Douglas, R. J. & Martin, K. A. in *Single neuron computation* 381–412 (Elsevier, 1992).

43. Douglas, R. J. & Martin, K. A. in *The visual system from genesis to maturity* 213–232 (Springer, 1992).

44. Gori, M. & Tesi, A. On the problem of local minima in backpropagation. *IEEE Transactions on Pattern Analysis & Machine Intelligence,* 76–86 (1992).

45. Horiuchi, T., Bair, W., Bishofberger, B., Lazzaro, J. & Koch, C. Computing motion using analog VLSI Chips: an experimental comparison among different approaches. *International Journal of Computer Vision* **8,** 203–216 (1992).

46. Krogh, A. & Hertz, J. A. *A simple weight decay can improve generalization* in *Advances in neural information processing systems* (1992), 950–957.

47. Mahowald, M. *VLSI analogs of neuronal visual processing: a synthesis of form and function* PhD thesis (Department of Computation and Neural Systems, California Institute of Technology, Pasadena, CA., 1992).

48. Troy, J. & Robson, J. Steady discharges of X and Y retinal ganglion cells of cat under photopic illuminance. *Visual neuroscience* **9,** 535–553 (1992).

49. Van Essen, D., Anderson, C. & Felleman, D. Information processing in the primate visual system - An integrated systems perspective. *Science* **255,** 419–423 (1992).

50. Bailey, C. H. & Kandel, E. R. Structural changes accompanying memory storage. *Annual review of physiology* **55,** 397–426 (1993).

51. O'Keefe, J. & Recce, M. L. Phase relationship between hippocampal place units and the EEG theta rhythm. *Hippocampus* **3,** 317–330 (1993).

52. Softky, W. R. & Koch, C. The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs. *Journal of Neuroscience* **13,** 334–350 (1993).

53. Hinton, G. E. & Zemel, R. S. *Autoencoders, minimum description length and Helmholtz free energy* in *Advances in neural information processing systems* (1994), 3–10.

54. Horiuchi, T. K., Bishofberger, B. & Koch, C. *An analog VLSI saccadic eye movement system* in *Advances in neural information processing systems* (1994), 582–589.

55. Mahowald, M. *An Analog VLSI System for Stereoscopic Vision* (Kluwer, Boston, MA, 1994).

56. Pavasović, A., Andreou, A. & Westgate, C. Characterization of subthreshold MOS mismatch in transistors for VLSI systems. *Journal of VLSI Signal Processing* **8,** 75–85 (July 1994).

57. Van Vreeswijk, C., Abbott, L. & Ermentrout, G. B. When inhibition not excitation synchronizes neural firing. *Journal of computational neuroscience* **1,** 313–321 (1994).

58. Buonomano, D. V. & Merzenich, M. M. Temporal information transformed into a spatial code by a neural network with realistic properties. *Science* **267,** 1028–1030 (1995).

59. Douglas, R., Koch, C., Mahowald, M., Martin, K. & Suarez, H. Recurrent Excitation in Neocortical Circuits. *Science* **269,** 981–985 (1995).

60. Kramer, J., Sarpeshkar, R. & Koch, C. *An Analog VLSI Velocity Sensor* in *Proc. IEEE Int. Symp. Circuit and Systems* (Seattle, WA, May 1995), 413–416.

61. Mainen, Z. & Sejnowski, T. Reliability of spike timing in neocortical neurons. *Science* **268,** 1503–1506 (1995).

62. Schöner, G., Dose, M. & Engels, C. Dynamics of behavior : theory and applications for autonomous robot architectures. *Robotics and autonomous systems* **16,** 213–245 (1995).

63. Gerstner, W., Kempter, R., van Hemmen, J. L. & Wagner, H. A neuronal learning rule for sub-millisecond temporal coding. *Nature* **383,** 76 (1996).

64. Kramer, J., Sarpeshkar, R. & Koch, C. *Analog VLSI motion discontinuity detectors for image segmentation* in *Proc. IEEE Int. Symp. Circuit and Systems* **II** (Atlanta, GA, May 1996), 620–623.

65. Pratt, L. & Jennings, B. in *Learning to learn* 19–43 (Springer, 1996).

66. Thorpe, S., Fize, D. & Marlot, C. Speed of Processing in the Human Visual System. *Nature* **381,** 520–2 (1996).

67. Tian, J.-R. & Lynch, J. C. Corticocortical input to the smooth and saccadic eye movement subregions of the frontal eye field in Cebus monkeys. *Journal of Neurophysiology* **76,** 2754–2771 (1996).

68. Holland, O. *Grey Walter: the pioneer of real artificial life* in *Proceedings of the 5th International Workshop on Artificial Life, MIT Press, Cambridge* (1997), 34–44.

69. Ilg, U. J. Slow eye movements. *Progress in neurobiology* **53,** 293–329 (1997).

70. Indiveri, G. Winner-take-all networks with lateral excitation. *Analog Integrated Circuits and Signal Processing* **13,** 185–193 (May 1997).

71. Indiveri, G. & Verschure, P. in *Proceedings Artificial Neural Networks-ICANN97: Lausanne, Switzerland* (eds W. Gerstner, A. G. M. H. & Nicoud, J.-D.) 811–816 (Springer Verlag, 1997).

72. Kramer, J., Sarpeshkar, R. & Koch, C. Pulse-Based Analog VLSI Velocity Sensors. *IEEE Transactions on Circuits and Systems II* **44,** 86–101 (Feb. 1997).

73. Maass, W. Networks of spiking neurons: the third generation of neural network models. *Neural networks* **10,** 1659–1671 (1997).

74. Maris, M. & Mahowald, M. A line following robot with intentional visual selection. *INNS/ENNS/KNNS Newsletter*. Appearing with Vol.10, Num.2 of Neural Networks (Mar. 1997).

75. Markram, H., Lübke, J., Frotscher, M. & Sakmann, B. Regulation of Synaptic Efficacy by Coincidence of Postsynaptic APs and EPSPs. *Science* **275,** 213–215 (1997).

76. Bi, G.-Q. & Poo, M.-M. Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic Cell Type. *Journal of Neuroscience* **18,** 10464–10472 (1998).

77. Ermentrout, B. Neural networks as spatio-temporal pattern-forming systems. *Reports on progress in physics* **353,** 353–430 (1998).

78. Haddad, H., Khatib, M., Lacroix, S. & Chatila, R. *Reactive Navigation in Outdoor Environments Using Potential Fields* in *Proceedings of the IEEE International Conference on Robotics and Automation* (1998), 1232–1237.

79. Markram, H., Pikus, D., Gupta, A. & Tsodyks, M. Potential for multiple mechanisms, phenomena and algorithms for synaptic plasticity at single synapses. *Neuropharmacology* **37,** 489–500 (1998).

80. O'Keefe, L. P. & Movshon, J. A. Processing of first-and second-order motion signals by neurons in area MT of the macaque monkey. *Visual neuroscience* **15,** 305–317 (1998).

81. Shadlen, M. N. & Newsome, W. T. The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. *Journal of neuroscience* **18,** 3870–3896 (1998).

82. Etienne-Cummings, R., Gruev, V. & Ghani, M. A. *VLSI implementation of motion centroid localization for autonomous navigation* in *Advances in Neural Information Processing Systems* (1999), 685–691.

83. Hahnloser, R., Douglas, R., Mahowald, M. & Hepp, K. Feedback interactions between neuronal pointers and maps for attentional processing. *Nature Neuroscience* **2,** 746–752 (1999).

84. Rao, R. P. & Ballard, D. H. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience* **2,** 79 (1999).

85. Abbott, L. & Nelson, S. Synaptic plasticity: taming the beast. *Nature Neuroscience* **3,** 1178–1183 (Nov. 2000).

86. Bicho, E., Mallet, P. & Schöner, G. Target Representation on an Autonomous Vehicle with Low-Level Sensors. *The International Journal of Robotics Research* **19,** 424–447 (2000).

87. Blanchard, M., Rind, F. & Verschure, P. Collision avoidance using a model of the locust LGMD neuron. *Robotics and autonomous systems* **30,** 17–38 (2000).

88. Brunel, N. Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *Journal of computational neuroscience* **8,** 183–208 (2000).

89. Hahnloser, R., Sarpeshkar, R., Mahowald, M., Douglas, R. & Seung, S. Digital selection and analog amplification co-exist in an electronic circuit inspired by neocortex. *Nature* **405,** 947–951 (2000).

90. Maass, W. On the computational power of winner-take-all. *Neural Computation* **12,** 2519–2535 (2000).

91. Pouget, A., Dayan, P. & Zemel, R. Information processing with population codes. *Nature Reviews Neuroscience* **1,** 125–132 (2000).

92. Song, S., Miller, K. & Abbot, L. Competitive Hebbian learning through spike-timing-dependent plasticity. *Nature Neuroscience* **3,** 919–926 (2000).

93. Brunel, N. & Wang, X. J. Effects of neuromodulation in a cortical network model of object working memory dominated by recurrent inhibition. *Journal of Computational Neuroscience* **11,** 63–85 (2001).

94. Dayan, P. & Abbott, L. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems* ISBN: 9780262541855 (MIT Press, Cambridge, MA, USA, 2001).

95. Dissanayake, M. G., Newman, P., Clark, S., Durrant-Whyte, H. F. & Csorba, M. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on robotics and automation* **17,** 229–241 (2001).

96. Iida, F. Goal-Directed Navigation of an Autonomous Flying Robot Using Biologicallz Inspired Cheal Vision. *ISR, Proceedings of the 32nd,* 19–21 (2001).

97. Indiveri, G. A Current-mode Hysteretic Winner-take-all Network, with Excitatory and Inhibitory Coupling. *Analog Integrated Circuits and Signal Processing* **28,** 279–291 (Sept. 2001).

98. Thorpe, S., Delorme, A., Rullen, R. V., *et al.* Spike-based strategies for rapid processing. *Neural networks* **14,** 715–725 (2001).

99. Bohte, S. M., Kok, J. N. & La Poutre, H. Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing* **48,** 17–37 (2002).

100. Gabbiani, F., Krapp, H. G., Koch, C. & Laurent, G. Multiplicative computation in a visual neuron sensitive to looming. *Nature* **420,** 320–324. ISSN: 0028-0836 (2002).

101. Indiveri, G., Oswald, P. & Kramer, J. *An adaptive visual tracking sensor with a hysteretic winner-take-all network* in *International Symposium on Circuits and Systems, (ISCAS), 2002* (May 2002), 324–327. doi:10.1109/ISCAS.2002.1010990.

102. Jaeger, H. *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach* (GMD-Forschungszentrum Informationstechnik, 2002).

103. Liu, S.-C., Kramer, J., Indiveri, G., Delbruck, T. & Douglas, R. *Analog VLSI:Circuits and Principles* (MIT Press, 2002).

104. Perrone, J. A. & Thiele, A. A model of speed tuning in MT neurons. *Vision research* **42,** 1035–1051 (2002).

105. Roberts, P. D. & Bell, C. C. Spike timing dependent synaptic plasticity in biological systems. *Biological cybernetics* **87,** 392–403 (2002).

106. Nishida, S., Sasaki, Y., Murakami, I., Watanabe, T. & Tootell, R. B. Neuroimaging of direction-selective mechanisms for second-order motion. *Journal of Neurophysiology* **90,** 3242–3254 (2003).

107. Wehr, M. & Zador, A. M. Balanced inhibition underlies tuning and sharpens spike timing in auditory cortex. *Nature* **426,** 442 (2003).

108. Binzegger, T., Douglas, R. & Martin, K. A Quantitative Map of the Circuit of Cat Primary Visual Cortex. *Journal of Neuroscience* **24,** 8441–53 (2004).

109. Cui, Z. *et al.* Inducible and reversible NR1 knockout reveals crucial role of the NMDA receptor in preserving remote memories in the brain. *Neuron* **41,** 781–793 (2004).

110. Douglas, R. J. & Martin, K. A. Neuronal circuits of the neocortex. *Annu. Rev. Neurosci.* **27,** 419–451 (2004).

111. Grill-Spector, K. & Malach, R. The Human Visual Cortex. *Annual Review of Neuroscience* **27,** 649–677 (2004).

112. Markram, H., Toledo-Rodriguez, M., *et al.* Interneurons of the neocortical inhibitory system. *Nature Reviews Neuroscience* **5,** 793–807 (2004).

113. Masland, R. H. Neuronal cell types. *Current Biology* **14,** R497–R500 (2004).

114. Mauk, M. D. & Buonomano, D. V. The neural basis of temporal processing. *Annu. Rev. Neurosci.* **27,** 307–340 (2004).

115. Rind, F. C. & Santer, R. D. Collision avoidance and a looming sensitive neuron: size matters but biggest is not necessarily best. *Proceedings. Biological sciences / The Royal Society* **271,** 27–9. ISSN: 0962-8452 (2004).

116. Yadid-Pecht, O. & Etienne-Cummings, R. *CMOS imagers* doi:10.1007/B117398 (Springer, 2004).

117. Abraham, W. C. & Robins, A. Memory retention–the synaptic stability versus plasticity dilemma. *Trends in neurosciences* **28,** 73–78 (2005).

118. Brette, R. & Gerstner, W. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *Journal of neurophysiology* **94,** 3637–3642 (2005).

119. Eliasmith, C. A Unified Approach to Building and Controlling Spiking Attractor Networks. *Neural Computation* **17,** 1276–1314 (2005).

120. Krauzlis, R. The control of voluntary eye movements: new perspectives. *The Neuroscientist* **11,** 124–137 (2005).

121. Mitra, R., Jadhav, S., McEwen, B. S., Vyas, A. & Chattarji, S. Stress duration modulates the spatiotemporal patterns of spine formation in the basolateral amygdala. *Proceedings of the National Academy of Sciences* **102,** 9371–9376 (2005).

122. Triesch, J. *Synergies between intrinsic and synaptic plasticity in individual model neurons* in *Advances in neural information processing systems* (2005), 1417–1424.

123. VanRullen, R., Guyonneau, R. & Thorpe, S. J. Spike times make sense. *Trends in neurosciences* **28,** 1–4 (2005).

124. Averbeck, B. B., Latham, P. E. & Pouget, A. Neural correlations, population coding and computation. *Nature reviews. Neuroscience* **7,** 358–66. ISSN: 1471-003X (2006).

125. Cudeiro, J. & Sillito, A. M. Looking back: corticothalamic feedback and early visual processing. *Trends in neurosciences* **29,** 298–306 (2006).

126. Erlhagen, W. & Bicho, E. The dynamic neural field approach to cognitive robotics. *Journal of neural engineering* **3,** R36–54. ISSN: 1741-2560 (Sept. 2006).

127. Gütig, R. & Sompolinsky, H. The tempotron: a neuron that learns spike timing–based decisions. *Nature Neuroscience* **9,** 420–428 (2006).

128. Hipp, J. *et al.* Texture signals in whisker vibrations. *Journal of neurophysiology* **95,** 1792–1799 (2006).

129. Indiveri, G., Chicca, E. & Douglas, R. A VLSI array of low-power spiking neurons and bistable synapses with spike–timing dependent plasticity. *IEEE Transactions on Neural Networks* **17,** 211–221 (Jan. 2006).

130. Lichtsteiner, P., Delbruck, T. & Posch, C. *A 100dB dynamic range high-speed dual-line optical transient sensor with asynchronous readout* in *International Symposium on Circuits and Systems, (ISCAS), 2006* (2006).

131. Lichtsteiner, P., Posch, C. & Delbruck, T. *A 128x128 120 dB 30 mW asynchronous vision sensor that responds to relative intensity change* in *2006 IEEE ISSCC Digest of Technical Papers* (Feb. 2006), 508–509.

132. Pfister, J.-P. & Gerstner, W. *Beyond pair-based STDP: A phenomenological rule for spike triplet and frequency effects* in *Advances in neural information processing systems* (2006), 1081–1088.

133. Priebe, N. J., Lisberger, S. G. & Movshon, J. A. Tuning for spatiotemporal frequency and speed in directionally selective neurons of macaque striate cortex. *Journal of Neuroscience* **26,** 2941–2950 (2006).

134. Bartolozzi, C. & Indiveri, G. Synaptic dynamics in analog VLSI. *Neural Computation* **19,** 2581–2603 (Oct. 2007).

135. Bengio, Y., Lamblin, P., Popovici, D. & Larochelle, H. *Greedy layer-wise training of deep networks* in *Advances in neural information processing systems* (2007), 153–160.

136. Calisi, D., Farinelli, A., Iocchi, L. & Nardi, D. Multi-objective exploration and search for autonomous rescue robots. *Journal of Field Robotics* **24,** 763–777 (2007).

137. Chan, V., Liu, S.-C. & van Schaik, A. AER EAR: A matched silicon cochlea pair with address event representation interface. *IEEE Transactions on Circuits and Systems I* **54** (eds Yadid-Pecht, O., Wilson, D. & Zaghloul, M.) Special Issue on Sensors, 48–59 (Jan. 2007).

138. Douglas, R. & Martin, K. Recurrent neuronal circuits in the neocortex. *Current Biology* **17,** R496–R500 (2007).

139. Douglas, R. & Martin, K. Mapping the Matrix: The Ways of Neocortex. *Neuron* **56,** 226–238. ISSN: 0896-6273 (2007).

140. Karmarkar, U. R. & Buonomano, D. V. Timing in the absence of clocks: encoding time in neural network states. *Neuron* **53,** 427–438 (2007).

141. Kilner, J. M., Friston, K. J. & Frith, C. D. Predictive coding: an account of the mirror neuron system. *Cognitive processing* **8,** 159–166 (2007).

142. Nvidia, C. Compute unified device architecture programming guide (2007).

143. Davison, A. *et al.* PyNN: a common interface for neuronal network simulators. *Frontiers in Neuroinformatics* **2,** 1–10 (2008).

144. Fasnacht, D., Whatley, A. & Indiveri, G. *A Serial Communication Infrastructure for Multi-Chip Address Event System* in *International Symposium on Circuits and Systems, (ISCAS), 2008* (May 2008), 648–651. doi:10.1109/ISCAS.2008.4541501.

145. Goodman, D. & Brette, R. Brian: a simulator for spiking neural networks in Python. *Frontiers in Neuroinformatics* **2.** doi:10.3389/neuro.01.026.2009 (2008).

146. Kole, M. H. *et al.* Action potential generation requires a high sodium channel density in the axon initial segment. *Nature neuroscience* **11,** 178 (2008).

147. Kurt, S. *et al.* Auditory cortical contrast enhancing by global winner-take-all inhibitory interactions. *PLoS ONE* **3** (2008).

148. Lichtsteiner, P., Posch, C. & Delbruck, T. A 128x128 120 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor. *IEEE Journal of Solid-State Circuits* **43,** 566–576. ISSN: 0018-9200 (Feb. 2008).

149. Theobald, J. C., Duistermars, B. J., Ringach, D. L. & Frye, M. A. Flies see second-order motion. *Current Biology* **18,** R464–R465 (2008).

150. Vincent, P., Larochelle, H., Bengio, Y. & Manzagol, P.-A. *Extracting and composing robust features with denoising autoencoders* in *Proceedings of the 25th international conference on Machine learning* (2008), 1096–1103.

151. Wolfe, J. *et al.* Texture coding in the rat whisker system: slip-stick versus differential resonance. *PLoS biology* **6,** e215 (2008).

152. Bartolozzi, C. & Indiveri, G. Global scaling of synaptic efficacy: Homeostasis in silicon synapses. *Neurocomputing* **72,** 726–731 (Jan. 2009).

153. Buonomano, D. V. & Maass, W. State-dependent computations: spatiotemporal processing in cortical networks. *Nature Reviews Neuroscience* **10,** 113–125 (2009).

154. Conradt, J., Cook, M., *et al. A pencil balancing robot using a pair of AER dynamic vision sensors* in *International Symposium on Circuits and Systems, (ISCAS), 2009* (May 2009), 781–784. doi:10.1109/ISCAS.2009.5117867.

155. Dean, P., Porrill, J., Ekerot, C.-F. & Jörntell, H. The cerebellar microcircuit as an adaptive filter: experimental and computational evidence. *Nature Reviews Neuroscience* **11 VN - re,** 30–43. ISSN: 1471-003X (2009).

156. Douglas, R. J. & Martin, K. A. Inhibition in cortical circuits. *Current Biology* **19,** R398–R402 (2009).

157. Goodman, D. & Brette, R. The Brian simulator. *Frontiers in Neuroscience* **3,** 192–197 (2009).

158. Holtmaat, A. & Svoboda, K. Experience-dependent structural synaptic plasticity in the mammalian brain. *Nature Reviews Neuroscience* **10,** 647 (2009).

159. Horiuchi, T. A Spike-Latency Model for Sonar-Based Navigation in Obstacle Fields. *Circuits and Systems I: Regular Papers, IEEE Transactions on* **56,** 2393–2401. ISSN: 1549-8328 (Nov. 2009).

160. Indiveri, G., Chicca, E. & Douglas, R. Artificial cognitive systems: From VLSI networks of spiking neurons to neuromorphic cognition. *Cognitive Computation* **1,** 119–127 (2009).

161. Ko, A. W. & Lau, H. Y. Intelligent robot-assisted humanitarian search and rescue system. *International Journal of Advanced Robotic Systems* **6,** 12 (2009).

162. Lukoševičius, M. & Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Computer Science Review* **3,** 127–149 (2009).

163. Mitra, S., Fusi, S. & Indiveri, G. Real-time classification of complex patterns using spike-based learning in neuromorphic VLSI. *Biomedical Circuits and Systems, IEEE Transactions on* **3,** 32–42 (Feb. 2009).

164. Oster, M., Douglas, R. & Liu, S.-C. Computation with Spikes in a Winner-Take-All Network. *Neural Computation* **21,** 2437–2465 (2009).

165. Quigley, M. *et al. ROS: an open-source Robot Operating System* in *ICRA Workshop on Open Source Software* (2009).

166. Restivo, L., Vetere, G., Bontempi, B. & Ammassari-Teule, M. The formation of recent and remote memory is associated with time-dependent formation of dendritic spines in the hippocampus and anterior cingulate cortex. *Journal of Neuroscience* **29,** 8206–8214 (2009).

167. Rutishauser, U. & Douglas, R. State-Dependent Computation Using Coupled Recurrent Networks. *Neural Computation* **21,** 478–509 (2009).

168. Vogels, T. P. & Abbott, L. Gating multiple signals through detailed balance of excitation and inhibition in spiking networks. *Nature neuroscience* **12,** 483 (2009).

169. Arel, I., Rose, D. C., Karnowski, T. P., *et al.* Deep machine learning-a new frontier in artificial intelligence research. *IEEE computational intelligence magazine* **5,** 13–18 (2010).

170. Liu, S., van Schaik, A., Minch, B. & Delbruck, T. *Event-based 64-channel binaural silicon cochlea with Q enhancement mechanisms* in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on* (2010), 2027–2030.

171. Liu, S.-C., Mesgarani, N., Harris, J. & Hermansky, H. *The use of spike-based representations for hardware audition systems* in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on* (2010), 505–508.

172. Nair, V. & Hinton, G. E. *Rectified linear units improve restricted boltzmann machines* in *Proceedings of the 27th international conference on machine learning (ICML-10)* (2010), 807–814.

173. Pfeiffer, M., Nessler, B., Douglas, R. J. & Maass, W. Reward-Modulated Hebbian Learning of Decision Making. *Neural Computation* **22,** 1399–1444 (Feb. 2010).

174. Posch, C., Matolin, D. & Wohlgenannt, R. *A QVGA 143 dB dynamic range asynchronous address-event PWM dynamic image sensor with lossless pixel-level video compression* in *International Solid-State Circuits Conference Digest of Technical Papers, ISSCC 2010* (Feb. 2010), 400–401. doi:10.1109/ISSCC.2010.5433973.

175. Posch, C., Matolin, D., Wohlgenannt, R., *et al. Live demonstration: Asynchronous time-based image sensor (ATIS) camera with full-custom AE processor* in *International Symposium on Circuits and Systems, (ISCAS), 2010* (Paris, France, 2010), 1392. doi:10.1109/ISCAS.2010.5537265.

176. Rahtu, E., Kannala, J., Salo, M. & Heikkilä, J. Segmenting salient objects from images and videos. *Computer Vision–ECCV 2010,* 366–379 (2010).

177. Roberts, P. D. & Leen, T. K. Anti-hebbian spike-timing-dependent plasticity and adaptive sensory processing. *Frontiers in computational neuroscience* **4,** 156 (2010).

178. Rokszin, A. *et al.* Visual pathways serving motion detection in the mammalian brain. *Sensors* **10,** 3218–3242 (2010).

179. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. & Manzagol, P.-A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research* **11,** 3371–3408 (2010).

180. Zingg, S., Scaramuzza, D., Weiss, S. & Siegwart, R. *MAV navigation through indoor corridors using optical flow* in *Robotics and Automation (ICRA), 2010 IEEE International Conference on* (2010), 3361–3368.

181. Benosman, R., Ieng, S.-H., Clercq, C., Bartolozzi, C. & Srinivasan, M. Asynchronous Frameless Event-based Optical Flow. *IEEE Transactions on Neural Natworks* (2011).

182. Bicho, E., Erlhagen, W., Louro, L. & Costa e Silva, E. Neuro-cognitive mechanisms of decision making in joint action: A human-robot interaction study. *Human Movement Science* **30,** 846–868. ISSN: 01679457 (2011).

183. Doyle, J. C. & Csete, M. Architecture, constraints, and behavior. *Proceedings of the National Academy of Sciences* **108,** 15624–15630 (2011).

184. Finger, H. & Liu, S.-C. *Estimating the location of a sound source with a spike-timing localization algorithm* in *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on* (2011), 2461–2464.

185. Gjorgjieva, J., Clopath, C., Audet, J. & Pfister, J.-P. A triplet spike-timing–dependent plasticity model generalizes the Bienenstock–Cooper–Munro rule to higher-order spatiotemporal correlations. *Proceedings of the National Academy of Sciences* **108,** 19383–19388 (2011).

186. Indiveri, G., Linares-Barranco, B., *et al.* Neuromorphic silicon neuron circuits. *Frontiers in Neuroscience* **5,** 1–23. ISSN: 1662-453X (2011).

187. Masci, J., Meier, U., Cireşan, D. & Schmidhuber, J. *Stacked convolutional auto-encoders for hierarchical feature extraction* in *International Conference on Artificial Neural Networks* (2011), 52–59.

188. Müller, G. R. & Conradt, J. A miniature low-power sensor system for real time 2D visual tracking of LED markers. *2011 IEEE International Conference on Robotics and Biomimetics, ROBIO 2011,* 2429–2434 (2011).

189. Neftci, E., Chicca, E., Indiveri, G. & Douglas, R. A systematic method for configuring VLSI networks of spiking neurons. *Neural Computation* **23,** 2457–2497 (Oct. 2011).

190. Oberlaender, M. *et al.* Cell type–specific three-dimensional structure of thalamocortical circuits in a column of rat vibrissal cortex. *Cerebral cortex* **22,** 2375–2391 (2011).

191. Reimann, H., Iossifidis, I. & Schöner, G. *Autonomous movement generation for manipulators with multiple simultaneous constraints using the attractor dynamics approach* in *IEEE International Conference on Robotics and Automation (ICRA)* (2011), 1050–4729.

192. Vogels, T. P., Sprekeler, H., Zenke, F., Clopath, C. & Gerstner, W. Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks. *Science* **334,** 1569–1573 (2011).

193. Bichler, O., Querlioz, D., Thorpe, S. J., Bourgoin, J.-P. & Gamrat, C. Extraction of temporally correlated features from dynamic vision sensors with spike-timing-dependent plasticity. *Neural Networks* **32,** 339–348 (2012).

194. Camunas-Mesa, L. *et al.* An event-driven multi-kernel convolution processor module for event-driven vision sensors. *IEEE Journal of Solid-State Circuits* **47,** 504–517 (2012).

195. Feldman, D. E. The spike-timing dependence of plasticity. *Neuron* **75,** 556–571 (2012).

196. Furber, S. B., Lester, D. R., *et al.* Overview of the SpiNNaker System Architecture. *IEEE Transactions on Computers* **62,** 2454–2467 (2012).

197. Graupner, M. & Brunel, N. Calcium-based plasticity model explains sensitivity of synaptic changes to spike pattern, rate, and dendritic location. *Proceedings of the National Academy of Sciences* **109,** 3991–3996 (2012).

198. Khansari-Zadeh, S. M. & Billard, A. A dynamical system approach to realtime obstacle avoidance. *Autonomous Robots* **32,** 433–454. ISSN: 0929-5593 (Mar. 2012).

199. Kim, K. *et al.* A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications. *Nano letters* **12,** 389–395 (2012).

200. Krizhevsky, A., Sutskever, I. & Hinton, G. *Imagenet classification with deep convolutional neural networks* in *Advances in neural information processing systems (NIPS)* (eds Pereira, F., Burges, C., Bottou, L. & Weinberger, K.) (Curran Associates, Inc., 2012), 1097–1105.

201. Kullmann, D. M., Moreau, A. W., Bakiri, Y. & Nicholson, E. Plasticity of inhibition. *Neuron* **75,** 951–962 (2012).

202. Neftci, E., Binas, J., Chicca, E., Indiveri, G. & Douglas, R. in *Biomimetic and Biohybrid Systems* (eds Prescott, T., Lepora, N., Mura, A. & Verschure, P.) 382–383 (Springer Berlin / Heidelberg, 2012). ISBN: 978-3-642-31524-4. doi:10.1007/978-3-642-31525-1_52.

203. Squire, L. *et al. Fundamental neuroscience* (Academic Press, 2012).

204. Thrun, S. & Pratt, L. *Learning to learn* (Springer Science & Business Media, 2012).

205. Baudot, P. *et al.* Animation of natural scene by virtual eye-movements evokes high precision and low noise in V1 neurons;handwritten. *Frontiers in neural circuits* **7,** 206 (2013).

206. Denk, C. *et al. Real-time interface board for closed-loop robotic tasks on the spinnaker neural computing system* in *International Conference on Artificial Neural Networks, ICANN 2013* **8131** (2013), 467–474.

207. Ercsey-Ravasz, M. *et al.* A predictive network model of cerebral cortical connectivity based on a distance rule. *Neuron* **80,** 184–197 (2013).

208. Hasler, J. & Marr, B. Finding a roadmap to achieve large neuromorphic hardware systems. *Frontiers in Neuroscience* **7,** 1–29 (Sept. 2013).

209. Hoffmann, R., Weikersdorfer, D. & Conradt, J. *Autonomous indoor exploration with an event-based visual SLAM system* in *Mobile Robots (ECMR), 2013 European Conference on* (2013), 38–43.

210. Laje, R. & Buonomano, D. V. Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nature neuroscience* **16,** 925 (2013).

211. Maisak, M. S. *et al.* A directional tuning map of Drosophila elementary motion detectors. *Nature* **500,** 212–216 (2013).

212. Mostafa, H., Müller, L. K. & Indiveri, G. *Recurrent networks of coupled Winner-Take-All oscillators for solving constraint satisfaction problems* in *Advances in Neural Information Processing Systems (NIPS)* (eds Burges, C., Bottou, L., Welling, M., Ghahramani, Z. & Weinberger, K.) **26** (2013), 719–727.

213. Perez-Peña, F., Morgado-Estevez, A., *et al.* Neuro-inspired spike-based motion: from dynamic vision sensor to robot motor open-loop control through spike-VITE. *Sensors (Basel, Switzerland)* **13,** 15805–15832. ISSN: 14248220 (2013).

214. Sandamirskaya, Y., Zibner, S. K., Schneegans, S. & Schöner, G. Using Dynamic Field Theory to extend the embodiment stance toward higher cognition. *New Ideas in Psychology* **31,** 322–339. ISSN: 0732-118X (2013).

215. Shin, H.-C., Orton, M. R., Collins, D. J., Doran, S. J. & Leach, M. O. Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data. *IEEE transactions on pattern analysis and machine intelligence* **35,** 1930–1943 (2013).

216. Weinzaepfel, P., Revaud, J., Harchaoui, Z. & Schmid, C. *DeepFlow: Large displacement optical flow with deep matching* in *Proceedings of the IEEE International Conference on Computer Vision* (2013), 1385–1392.

217. Benjamin, B. V. *et al.* Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations. *Proceedings of the IEEE* **102,** 699–716 (2014).

218. Benosman, R., Clercq, C., Lagorce, X., Ieng, S.-H. & Bartolozzi, C. Event-based visual flow. *Neural Networks and Learning Systems, IEEE Transactions on* **25,** 407–417 (2014).

219. Brandli, C., Berner, R., Yang, M., Liu, S.-C. & Delbruck, T. A 240×180 130 dB 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits* **49,** 2333–2341. ISSN: 0018-9200 (2014).

220. Censi, A. & Scaramuzza, D. *Low-latency event-based visual odometry* in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (2014), 703–710.

221. Chicca, E., Stefanini, F., Bartolozzi, C. & Indiveri, G. Neuromorphic electronic circuits for building autonomous cognitive systems. *Proceedings of the IEEE* **102,** 1367–1388. ISSN: 0018-9219 (Sept. 2014).

222. Clady, X. *et al.* Asynchronous Visual Event-based Time-to-Contact. *Frontiers in Neuroscience* **8.** ISSN: 1662-453X. doi:10.3389/fnins.2014.00009 (2014).

223. Egelhaaf, M., Kern, R. & Lindemann, J. P. Motion as a source of environmental information: a fresh view on biological motion computation by insect brains. *Frontiers in neural circuits* **8** (2014).

224. Fu, C., Olivares-Mendez, M. A., Suarez-Fernandez, R. & Campoy, P. Monocular visual-inertial slam-based collision avoidance strategy for fail-safe uav using fuzzy logic controllers. *Journal of Intelligent & Robotic Systems* **73,** 513–533 (2014).

225. Furber, S., Galluppi, F., Temple, S. & Plana, L. The SpiNNaker Project. *Proceedings of the IEEE* **102,** 652–665. ISSN: 0018-9219 (May 2014).

226. Galluppi, F. *et al. Event-based neural computing on an autonomous mobile platform* in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (2014), 2862–2867.

227. Girshick, R., Donahue, J., Darrell, T. & Malik, J. *Rich feature hierarchies for accurate object detection and semantic segmentation* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2014), 580–587.

228. Hangya, B., Pi, H.-J., Kvitsiani, D., Ranade, S. P. & Kepecs, A. From circuit motifs to computations: mapping the behavioral repertoire of cortical interneurons. *Current opinion in neurobiology* **26,** 117–124 (2014).

229. Jia, Y. *et al. Caffe: Convolutional Architecture for Fast Feature Embedding* in *Proceedings of the 22Nd ACM International Conference on Multimedia* (ACM, Orlando, Florida, USA, 2014), 675–678. ISBN: 978-1-4503-3063-3. doi:10.1145/2647868.2654889. http://doi.acm.org/10.1145/2647868.2654889.

230. Kappel, D., Nessler, B. & Maass, W. STDP installs in winner-take-all circuits an online approximation to hidden Markov model learning. *PLoS computational biology* **10,** e1003511 (2014).

231. Kepecs, A. & Fishell, G. Interneuron cell types are fit to function. *Nature* **505,** 318 (2014).

232. Koziol, S., Brink, S. & Hasler, J. A neuromorphic approach to path planning using a reconfigurable neuron array IC. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **22,** 2724–2737. ISSN: 10638210 (2014).

233. Liu, S.-C., Delbruck, T., Indiveri, G., Whatley, A. & Douglas, R. *Event-based neuromorphic systems* doi:10.1002/9781118927601.ch6 (Wiley, 2014).

234. Manen, S., Kwon, J., Guillaumin, M. & Van Gool, L. *Appearances can be deceiving: Learning visual tracking from few trajectory annotations* in *European Conference on Computer Vision, ECCV 2014* **8693** (2014), 157–172.

235. Mauss, A. S., Meier, M., Serbe, E. & Borst, A. Optogenetic and pharmacologic dissection of feedforward inhibition in Drosophila motion vision. *Journal of Neuroscience* **34,** 2254–2263 (2014).

236. Muir, D. R. & Cook, M. Anatomical constraints on lateral competition in columnar cortical architectures. *Neural computation* **26,** 1624–1666 (2014).

237. Patel, G. H., Kaplan, D. M. & Snyder, L. H. Topographic organization in the brain: searching for general principles. *Trends in cognitive sciences* **18,** 351–363 (2014).

238. Patterson, D. A. & Hennessy, J. L. *Computer Organization and Design - The Hardware/Software Interface* (Elsevier, 2014).

239. Posch, C., Serrano-Gotarredona, T., Linares-Barranco, B. & Delbruck, T. Retinomorphic event-based vision sensors: bioinspired cameras with spiking output. *Proceedings of the IEEE* **102,** 1470–1484 (2014).

240. Ramachandran, H., Weber, S., Aamir, S. A. & Chicca, E. *Neuromorphic Circuits for Short-term Plasticity with Recovery Control* in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)* (2014), 858–861. doi:10.1109/ISCAS.2014.6865271.

241. Sandamirskaya, Y. Dynamic neural fields as a step toward cognitive neuromorphic architectures. *Frontiers in Neuroscience* **7** (2014).

242. Stefanini, F., Sheik, S., Neftci, E. & Indiveri, G. PyNCS: a microkernel for high-level definition and configuration of neuromorphic electronic systems. *Frontiers in Neuroinformatics* **8.** doi:10.3389/fninf.2014.00073 (2014).

243. Truong, S. N., Ham, S.-J. & Min, K.-S. Neuromorphic crossbar circuit with nanoscale filamentary-switching binary memristors for speech recognition. *Nanoscale Research Letters* **9,** 629. ISSN: 1556-276X (Nov. 2014).

244. Akopyan, F. *et al.* Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **34,** 1537–1557 (2015).

245. Bartol Jr, T. M. *et al.* Nanoconnectomic upper bound on the variability of synaptic plasticity. *Elife* **4,** e10778 (2015).

246. Bengio, Y., Lee, D.-H., Bornschein, J., Mesnard, T. & Lin, Z. Towards biologically plausible deep learning. *arXiv preprint arXiv:1502.04156* (2015).

247. Bertrand, O. J. N., Lindemann, J. P. & Egelhaaf, M. A bio-insprired collision avoidance model based on spatial information derived from motion detectors leads to common routes. *PLoS computational biology* **11,** e1004339 (2015).

248. Borst, A. & Helmstaedter, M. Common circuit design in fly and mammalian motion vision. *Nature Neuroscience* **18,** 1067–1076 (2015).

249. Brosch, T., Tschechne, S. & Neumann, H. On event-based optical flow detection. *Frontiers in Neuroscience* **9** (2015).

250. Burbank, K. S. Mirrored STDP implements autoencoder learning in a network of spiking neurons. *PLoS computational biology* **11,** e1004566 (2015).

251. Conradt, J. *On-board real-time optic-flow for miniature event-based vision sensors* in *Robotics and Biomimetics (ROBIO), 2015 IEEE International Conference on* (2015), 1858–1863.

252. Conradt, J., Galluppi, F. & Stewart, T. C. Trainable sensorimotor mapping in a neuromorphic robot. *Robotics and Autonomous Systems* **71,** 60–68. ISSN: 0921-8890 (2015).

253. Conti, F., Rossi, D., Pullini, A., Loi, I. & Benini, L. PULP: A Ultra-Low Power Parallel Accelerator for Energy-Efficient and Flexible Embedded Vision. *Journal of Signal Processing Systems,* 1–16 (2015).

254. Courbariaux, M., Bengio, Y. & David, J.-P. *Binaryconnect: Training deep neural networks with binary weights during propagations* in *Advances in neural information processing systems* (2015), 3123–3131.

255. George, R., Mayr, C., Indiveri, G. & Vassanelli, S. *Event-based softcore processor in a biohybrid setup applied to structural plasticity* in *Event-based Control, Communication, and Signal Processing (EBCCSP), 2015 International Conference on* (June 2015), 1–4. doi:10.1109/EBCCSP.2015.7300664.

256. Girshick, R. *Fast r-cnn* in *Proceedings of the IEEE international conference on computer vision* (2015), 1440–1448.

257. Goltstein, P. M., Montijn, J. S. & Pennartz, C. M. Effects of isoflurane anesthesia on ensemble patterns of Ca2+ activity in mouse v1: reduced direction selectivity independent of increased correlations in cellular activity. *PLoS One* **10,** e0118277 (2015).

258. Indiveri, G. & Liu, S.-C. Memory and information processing in neuromorphic systems. *Proceedings of the IEEE* **103,** 1379–1397 (2015).

259. Jiang, X. *et al.* Principles of connectivity among morphologically defined cell types in adult neocortex. *Science* **350,** aac9462 (2015).

260. Jordan, M. I. & Mitchell, T. M. Machine learning: Trends, perspectives, and prospects. *Science* **349,** 255–260 (2015).

261. Klein, P., Conradt, J. & Liu, S.-C. *Scene stitching with event-driven sensors on a robot head platform* in *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on* (2015), 2421–2424.

262. Lagorce, X., Ieng, S. H., Clady, X., Pfeiffer, M. & Benosman, R. B. Spatiotemporal features for asynchronous event-based data. *Frontiers in Neuroscience* **9,** 1–13. ISSN: 1662453X (2015).

263. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521,** 436–444 (2015).

264. Long, J., Shelhamer, E. & Darrell, T. *Fully convolutional networks for semantic segmentation* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), 3431–3440.

265. Luber, D., Biedermann, J. & Conradt, J. Chain of Small Robots. *Project Laboratory Computational Neuro Engineering* (2015).

266. Martel, J. N., Chau, M., Dudek, P. & Cook, M. *Toward joint approximate inference of visual quantities on cellular processor arrays* in *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on* (2015), 2061–2064.

267. Milde, M. B., Bertrand, O. J. N., Benosman, R., Egelhaaf, M. & Chicca, E. *Bioinspired event-driven collision avoidance algorithm based on optic flow* in *2015 International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)* (2015), 1–7. doi:10.1109/EBCCSP.2015.7300673.

268. Qiao, N. *et al.* A Re-configurable On-line Learning Spiking Neuromorphic Processor comprising 256 neurons and 128K synapses. *Frontiers in Neuroscience* **9,** 1–17. ISSN: 1662-453X (2015).

269. Ren, S., He, K., Girshick, R. & Sun, J. *Faster r-cnn: Towards real-time object detection with region proposal networks* in *Advances in neural information processing systems* (2015), 91–99.

270. *Dynamic Thinking: A Primer on Dynamic Field Theory* (eds Schöner, G. & Spencer, J.) (Oxford University Press, 2015).

271. Stromatias, E., Neil, D., *et al.* Robustness of spiking Deep Belief Networks to noise and reduced bit precision of neuro-inspired hardware platforms. *Frontiers in neuroscience* **9** (2015).

272. Sussillo, D., Churchland, M. M., Kaufman, M. T. & Shenoy, K. V. A neural network that finds a naturalistic solution for the production of muscle activity. *Nature neuroscience* **18,** 1025 (2015).

273. Yang, M., Liu, S.-C. & Delbruck, T. A dynamic vision sensor with 1% temporal contrast sensitivity and in-pixel asynchronous delta modulator for event encoding. *IEEE Journal of Solid-State Circuits* **50,** 2149–2160 (2015).

274. Zenke, F., Agnes, E. J. & Gerstner, W. Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks. *Nature Communications* **6,** 1–13 (Apr. 2015).

275. Bojarski, M. *et al.* End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316* (2016).

276. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. & Yuille, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915* (2016).

277. Deneve, S. & Machens, C. K. Efficient codes and balanced networks. *Nature neuroscience* **19,** 375 (2016).

278. Faessler, M. *et al.* Autonomous, Vision-based Flight and Live Dense 3D Mapping with a Quadrotor Micro Aerial Vehicle. *Journal of Field Robotics* **33,** 431–450 (2016).

279. Giulioni, M., Lagorce, X., Galluppi, F. & Benosman, R. Event-based computation of motion flow on a neuromorphic analog neural platform. *Frontiers in neuroscience* **10** (2016).

280. Goel, A. & Buonomano, D. V. Temporal interval learning in cortical cultures is encoded in intrinsic network dynamics. *Neuron* **91,** 320–327 (2016).

281. Goodfellow, I., Bengio, Y. & Courville, A. *Deep learning* (MIT press, 2016).

282. He, K., Zhang, X., Ren, S. & Sun, J. *Deep residual learning for image recognition* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 770–778.

283. Lillicrap, T. P., Cownden, D., Tweed, D. B. & Akerman, C. J. Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications* **7,** 13276 (2016).

284. Lotter, W., Kreiman, G. & Cox, D. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104* (2016).

285. Martín Abadi *et al.* Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).

286. Meyer, H. G. *et al. A bio-inspired model for visual collision avoidance on a hexapod walking robot* in *Conference on Biomimetic and Biohybrid Systems* (2016), 167–178.

287. Milde, M. B., Sandamirskaya, Y. & Indiveri, G. *Neurally-inspired robotic controllers implemented on neuromorphic hardware* in *Conference of Biomimetics* (Bremen, 2016).

288. Moeys, D. P. *et al. Steering a Predator Robot using a Mixed Frame / Event-Driven Convolutional Neural Network* in *Event-based Control, Communication, and Signal Processing (EBCCSP), 2016 Second International Conference on. IEEE* (2016), 1–8.

289. Mostafa, H., Mayr, C. & Indiveri, G. *Beyond spike-timing dependent plasticity in memristor crossbar arrays* in *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on* (2016), 926–929.

290. Neil, D., Pfeiffer, M. & Liu, S.-C. *Phased lstm: Accelerating recurrent network training for long or event-based sequences* in *Advances in Neural Information Processing Systems* (2016), 3882–3890.

291. Nvidia. https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/tesla-product-literature/NV-tesla-p100-pcie-PB-08248-001-v01.pdf (2016).

292. Robertson, C. E., Hermann, K. L., Mynick, A., Kravitz, D. J. & Kanwisher, N. Neural representations integrate the current field of view with the remembered 360 panorama in scene-selective cortex. *Current Biology* **26,** 2463–2468 (2016).

293. Rueckauer, B. & Delbruck, T. Evaluation of event-based algorithms for optical flow with ground-truth from inertial measurement sensor. *Frontiers in neuroscience* **10** (2016).

294. Spiess, R., George, R., Cook, M. & Diehl, P. U. Structural plasticity denoises responses and improves learning speed. *Frontiers in computational neuroscience* **10,** 93 (2016).

295. Stewart, T. C., Kleinhans, A., Mundy, A. & Conradt, J. Serendipitous Offline Learning in a Neuromorphic Robot. *Frontiers in Neurorobotics* **10,** 1–11. ISSN: 1662-5218 (2016).

296. Zhang, C., Bengio, S., Hardt, M., Recht, B. & Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530* (2016).

297. Zidan, M. *et al.* Single-readout high-density memristor crossbar. *Scientific reports* **6,** 18863 (2016).

298. Blum, H. *et al.* A neuromorphic controller for a robotic vehicle equipped with a dynamic vision sensor. *Robotics: Science and Systems, Proceedings of RSS 2017* (2017).

299. Cavigelli, L. & Benini, L. Origami: A 803-gop/s/w convolutional network accelerator. *IEEE Transactions on Circuits and Systems for Video Technology* **27,** 2461–2475 (2017).

300. George, R. *Structural plasticity in neuromorphic systems* PhD thesis (University of Zurich, 2017). doi:10.5167/uzh-153150.

301. Hwu, T., Isbell, J., Oros, N. & Krichmar, J. *A self-driving robot using deep convolutional neural networks on neuromorphic hardware* in *Neural Networks (IJCNN), 2017 International Joint Conference on* (2017), 635–641.

302. Hwu, T., Krichmar, J. & Zou, X. *A complete neuromorphic solution to outdoor navigation and path planning* in *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on* (2017), 1–4.

303. Lagorce, X., Orchard, G., Galluppi, F., Shi, B. E. & Benosman, R. B. Hots: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE transactions on pattern analysis and machine intelligence* **39,** 1346–1359 (2017).

304. Leinweber, M., Ward, D. R., Sobczak, J. M., Attinger, A. & Keller, G. B. A sensorimotor circuit in mouse cortex for visual flow predictions. *Neuron* **95,** 1420–1432 (2017).

305. Liu, M. & Delbruck, T. Block-matching optical flow for dynamic vision sensors: Algorithm and FPGA implementation, 1–4 (2017).

306. Lungu, I.-A., Corradi, F. & Delbrück, T. *Live demonstration: Convolutional neural network driven by dynamic vision sensor playing RoShamBo* in *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on* (2017), 1–1.

307. Milde, M. B., Neil, D., Aimar, A., Delbruck, T. & Indiveri, G. ADaPTION: Toolbox and Benchmark for Training Convolutional Neural Networks with Reduced Numerical Precision Weights and Activation. *ArXiv e-prints* (2017).

308. Milde, M. B., Blum, H., *et al.* Obstacle Avoidance and Target Acquisition for Robot Navigation Using a Mixed Signal Analog/Digital Neuromorphic Processing System. *Frontiers in Neurorobotics* **11,** 28. ISSN: 1662-5218 (2017).

309. Milde, M. B., Dietmüller, A., Blum, H., Indiveri, G. & Sandamirskaya, Y. *Obstacle avoidance and target acquisition in mobile robots equipped with neuromorphic sensory-processing systems* in *International Symposium on Circuits and Systems, (ISCAS), 2017* (2017). doi:10.1109/ISCAS.2017.8050976.

310. Neftci, E. O., Augustine, C., Paul, S. & Detorakis, G. Event-driven random back-propagation: Enabling neuromorphic deep learning machines. *Frontiers in neuroscience* **11,** 324 (2017).

311. Nozaki, Y. & Delbruck, T. Temperature and Parasitic Photocurrent Effects in Dynamic Vision Sensors. *IEEE Transactions on Electron Devices* **64,** 3239–3245 (2017).

312. Partzsch, J. *et al. A Fixed Point Exponential Function Accelerator for a Neuromorphic Many-Core System* in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)* (2017), 1091–1094.

313. Perez-Peña, F., Leñero-Bardallo, J. A., Linares-Barranco, A. & Chicca, E. *Towards bioinspired close-loop local motor control: a simulated approach supporting neuromorphic implementations* in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)* (2017).

314. Rueckauer, B., Lungu, I.-A., Hu, Y., Pfeiffer, M. & Liu, S.-C. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in neuroscience* **11,** 682 (2017).

315. Salt, L., Indiveri, G. & Sandamirskaya, Y. *Obstacle avoidance with LGMD neuron: towards a neuromorphic UAV implementation* in *International Symposium on Circuits and Systems (ISCAS), 2017* (2017), 1–4.

316. Serres, J. R. & Ruffier, R. Optic flow-based collision-free strategies: From insects to robots. *Arthropod Structure & Development* **46,** 703–717 (2017).

317. Shmelkov, K., Schmid, C. & Alahari, K. Incremental learning of object detectors without catastrophic forgetting. *arXiv preprint arXiv:1708.06977* (2017).

318. Soldado-Magraner, S. *et al.* Conditioning by subthreshold synaptic input changes the characteristic firing pattern of CA3 hippocampal neurons. *bioRxiv,* 084152 (2017).

319. Son, B. *et al. 4.1 A 640× 480 dynamic vision sensor with a 9μm pixel and 300Meps address-event representation* in *Solid-State Circuits Conference (ISSCC), 2017 IEEE International* (2017), 66–67.

320. Stromatias, E., Soto, M., Serrano-Gotarredona, T. & Linares-Barranco, B. An event-driven classifier for spiking neural networks fed with synthetic or dynamic vision sensor data. *Frontiers in neuroscience* **11,** 350 (2017).

321. Aimar, A. *et al.* Nullhop: A flexible convolutional neural network accelerator based on sparse representations of feature maps. *IEEE transactions on neural networks and learning systems,* 1–13 (2018).

322. Davies, M. *et al.* Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro* **38,** 82–99. ISSN: 0272-1732 (Jan. 2018).

323. Dos Santos, L. A. *Artificial Intelligence* https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/ (GitBooks, 2018).

324. Gallego, G., Rebecq, H. & Scaramuzza, D. *A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation* in *IEEE Int. Conf. Comput. Vis. Pattern Recog.(CVPR)* **1** (2018).

325. Gysel, P., Pimentel, J., Motamedi, M. & Ghiasi, S. Ristretto: A framework for empirical study of resource-efficient inference in convolutional neural networks. *IEEE Transactions on Neural Networks and Learning Systems* (2018).

326. Keller, G. B. & Mrsic-Flogel, T. D. Predictive processing: A canonical cortical computation. *Neuron* **100,** 424–435 (2018).

327. Liu, C., Bellec, G., *et al.* Memory-efficient Deep Learning on a SpiNNaker 2 prototype. *Frontiers in neuroscience* **12** (2018).

328. Milde, M. B., Bertrand, O. J. N., Ramachandran, H., Egelhaaf, M. & Chicca, E. Spiking Elementary Motion Detector in Neuromorphic Systems. *Neural computation* **30,** 2384–2417 (2018).

329. Moradi, S., Qiao, N., Stefanini, F. & Indiveri, G. A Scalable Multicore Architecture With Heterogeneous Memory Structures for Dynamic Neuromorphic Asynchronous Processors (DYNAPs). *Biomedical Circuits and Systems, IEEE Transactions on* **12,** 106–122 (Feb. 2018).

330. Müller, M. M., Bertrand, O. J., Differt, D. & Egelhaaf, M. The problem of home choice in skyline-based homing. *PloS one* **13,** e0194070 (2018).

331. Payvand, M., Muller, L. K. & Indiveri, G. *Event-based circuits for controlling stochastic learning with memristive devices in neuromorphic architectures* in *Circuits and Systems (ISCAS), 2018 IEEE International Symposium on* (2018), 1–5.

332. Payvand, M., Nair, M., Müller, L. & Indiveri, G. A neuromorphic systems approach to in-memory computing with non-ideal memristive devices: From mitigation to exploitation. *Faraday Discussions,* 1–13 (2018).

333. Remington, E. D., Narain, D., Hosseini, E. A. & Jazayeri, M. Flexible sensorimotor computations through rapid reconfiguration of cortical dynamics. *Neuron* **98,** 1005–1019 (2018).

334. Stroud, J. P., Porter, M. A., Hennequin, G. & Vogels, T. P. Motor primitives in space and time via targeted gain modulation in cortical networks. *Nature neuroscience* **21,** 1774 (2018).

335. Thakur, C. S. *et al.* Large-Scale Neuromorphic Spiking Array Processors: A Quest to Mimic the Brain. *Frontiers in Neuroscience* **12,** 891 (2018).

336. Thiele, J. C., Bichler, O. & Dupret, A. Event-based, timescale invariant unsupervised online deep learning with STDP. *Frontiers in Computational Neuroscience* **12,** 46 (2018).

337. Wang, R. & van Schaik, A. Breaking Liebig's Law: An Advanced Multipurpose Neuromorphic Engine. *Frontiers in neuroscience* **12** (2018).

338. Yousefzadeh, A., Orchard, G., Stromatias, E., Serrano-Gotarredona, T. & Linares-Barranco, B. *Hybrid Neural Network, An Efficient Low-Power Digital Hardware Implementation of Event-based Artificial Neural Network* in *Circuits and Systems (ISCAS), 2018 IEEE International Symposium on* (2018), 1–5.

339. Afshar, S., Tapson, J., van Schaik, A. & Cohen, G. FEAST: Feature Extraction using Adaptive Selection Thresholds (2019).

340. Bichler, O., Briand, D., *et al.* N2D2 - Neural Network Design & Deployment. *Manual available on Github* (2019).

341. Neckar, A. *et al.* Braindrop: A Mixed-Signal Neuromorphic Architecture With a Dynamical Systems-Based Programming Model. *Proceedings of the IEEE* **107,** 144–164 (2019).

342. *iniLabs Ltd* http://www.inilabs.com.

343. *The jAER Open Source Project* Github web-site. Nov. 2007. https://github.com/SensorsINI/jaer.

344. *Raggedstone2 with Spartan 6 FPGA* https://www.enterpoint.co.uk/products/spartan-6-development-boards/raggedstone-2/. 2017.

PART V

Appendix

7.1 Reactive control

7.1.1 Appendix A: Neuronal equations

The dynamics of the spiking neurons on the ROLLS chip can be approximated by the differential equation Eqs. (7.1-7.3), obtained by performing circuit analysis:

$$\tau \frac{dImem}{dt} = \frac{\frac{I_{th}}{I_\tau}(I_{in} - I_{ahp} - I_\tau) + \frac{I_a}{I_\tau}(I_{mem} + I_{th}) - I_{mem}(1 + \frac{I_{ahp}}{I_\tau})}{(1 + \frac{I_{th}}{I_{mem}})}, \tag{7.1}$$

$$\tau_{ahp} \frac{dI_{ahp}}{dt} = \frac{I_{thahp}}{I_{\tau ahp}} I_{Ca} u(t) - I_{ahp}, \tag{7.2}$$

where I_{mem} is the membrane potential, I_{ahp} is the adaptation current, $u(t)$ is a step function that is one during spikes and zero otherwise, I_τ and $I_{\tau ahp}$ are time constant currents, $I_t h$ and I_{thahp} are currents through N-type MOSFETs, τ and τ_{ahp} are time constants, and I_{in} is the input current from the synapses.

The time constants are dependent on the time constant currents and can be calculated by:

$$\tau = \frac{C_{mem} U_T}{\kappa I_{tau}}, \tag{7.3}$$

where κ is a MOSFET property, U_T is the thermal potential, and C_{mem} is the membrane capacitance. τ_{ahp} is calculated similarly except it substitutes I_τ with $I_{\tau ahp}$ and C_{mem} with C_p.

These equations approximate an adaptive integrate-and-fire dynamics (Brette & Gerstner, 2005) .

7.1.2 Appendix B: Biases of the ROLLS chip used in our experiments

Table 7.1 shows the biases used for our experiments to set-up non-plastic connections between the neuronal populations; Table 7.2 shows biases for the integrate-and-fire neurons on chip. Each bias corresponds to a current, supplied to the neuronal circuits and is calculated as Range × Value, where letters near the range mean the order of magnitude: “p” – piko, “n” – nano, “u” – micro (see (Qiao *et al.,* 2015) for details of the circuit and functional meaning of the biases). The biases are set using software and FPGA-mapping, implemented on the Parallella board.

The eight different values of the synaptic weights that we used in our architecture (-4 : 4) are obtained combining the NPA_WEIGHT_INH_N, NPA_WEIGHT_INH_N1, and NPA_WEIGHT_INH_N2 biases for negative weights and the NPA_WEIGHT_EXC_P,

Table 7.1: Hardware biases for the non-plastic synapses

Bias name	Range (A)	Value	Flags
NPA_PWLK_P	820p	200	H
NPA_WEIGHT_STD_N	15p	15	H N
NPA_WEIGHT_EXC_P	820p	123	H
NPA_WEIGHT_EXC0_P	0.4u	15	H
NPA_WEIGHT_EXC1_P	0.4u	82	H
NPDPIE_THR_P	820p	38	H
NPDPIE_TAU_P	105p	22	H
NPA_WEIGHT_INH_N	820p	82	H N
NPA_WEIGHT_INH_N0	820p	200	H N
NPA_WEIGHT_INH_N1	6.5n	71	H N
NPDPII_TAU_P	15p	51	H
NPDPII_THR_P	820p	177	H

NPA_WEIGHT_EXC_P1, and NPA_WEIGHT_EXC_P2 biases for positive weights:

$$1 = \text{NPA\_WEIGHT\_EXC\_P}$$

$$2 = \text{NPA\_WEIGHT\_EXC\_P} + \text{NPA\_WEIGHT\_EXC\_P1}$$

$$3 = \text{NPA\_WEIGHT\_EXC\_P} + \text{NPA\_WEIGHT\_EXC\_P2}$$

$$4 = \text{NPA\_WEIGHT\_EXC\_P} + \text{NPA\_WEIGHT\_EXC\_P1} + \text{NPA\_WEIGHT\_EXC\_P2}$$

$$-1 = \text{NPA\_WEIGHT\_INH\_N}$$

$$-2 = \text{NPA\_WEIGHT\_INH\_N} + \text{NPA\_WEIGHT\_INH\_N1}$$

$$-3 = \text{NPA\_WEIGHT\_INH\_N} + \text{NPA\_WEIGHT\_INH\_N2}$$

$$-4 = \text{NPA\_WEIGHT\_INH\_N} + \text{NPA\_WEIGHT\_INH\_N1} + \text{NPA\_WEIGHT\_INH\_N2}$$

Table 7.2: Hardware biases for integrate-and-fire neurons

Bias name	Range (A)	Value	Flags
IF_RST_N	15p	17	H N
IF_BUR_P	50p	56	H N
IF_ATHR_N	15p	0	H N
IF_RFR1_N	820p	50	H N
IF_RFR2_N	820p	50	H N
IF_AHW_P	15p	0	H
IF_AHTAU_N	820p	37	N
IF_DC_P	15p	0	H
IF_TAU2_N	105p	77	N
IF_TAU1_N	105p	100	N
IF_NMDA_N	15p	17	H N
IF_CASC_N	15p	17	H N
IF_THR_N	820p	100	H N

7.2 A sense of motion and distance

7.2.1 Robotic setup and outdoor cluttered environment

The DVS and a standard webcam (Logitech Webcam 200) were mounted on a pioneer 2-DX robot. Both the DVS and the webcam were facing in the direction of motion of the robot (Fig. 7.2 left). This system was used to acquire data in outdoor conditions with 8 cylindrical objects ($\Phi \times \mathrm{H} = 9 \times 74\ cm$), covered with a 9 mm black and white chequerboard pattern (Fig. 7.1, 7.2 right). The robot moved along a straight line at 0.7 ms^{-1} and was manually controlled. Data were recorded using a Python implementation in Robot Operating System (ROS) Indigo (Quigley *et al.,* 2009), which interfaced the DVS using the jAER graphical user interface (*The jAER Open Source Project* 2007). The jAER viewer forwarded the raw data via a user datagram protocol (UDP) to ROS. Python 2.7 was used to read out the files published by ROS.

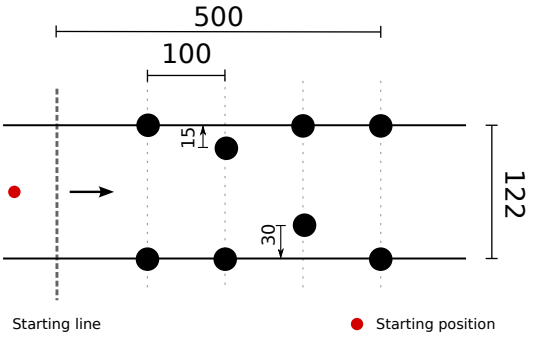


Figure 7.1: Object constellation formed by 8 cylindrical objects (black dots) in open field. The robot started from the starting position (red dot). The distance between each object along the trajectory is 100 cm. In total the robot travelled 500cm. The distance of each object to the white lines (Fig. 7.2 right), varies between 0 and 30 cm.

Figure 7.2: Robotic setup and outdoor environment. Left: The DVS and a standard webcam mounted on the Pioneer 2-DX robot Right: Experimental setup of 8 objects in outdoor condition, covered with black/white chequerboard pattern.

7.2.2 Spiking elementary motion detector on SpiNNaker

To investigate the closed-loop behaviour of the spiking Elementary Motion Detector (sEMD) in future studies we implemented the sEMD on a digital neuromorphic processor. We implemented the sEMD on a SpiNNaker board (Furber, Galluppi, *et al.,* 2014) using PyNN (Davison *et al.,* 2008). The SpiNNaker board is connected to a eDVS (Müller & Conradt, 2011) using the SpiNNaker IO-board (Denk *et al.,* 2013) kindly provided by the Neuroscientific System Theory group of Jörg Conradt. Fig. 7.3 shows the tuning curve

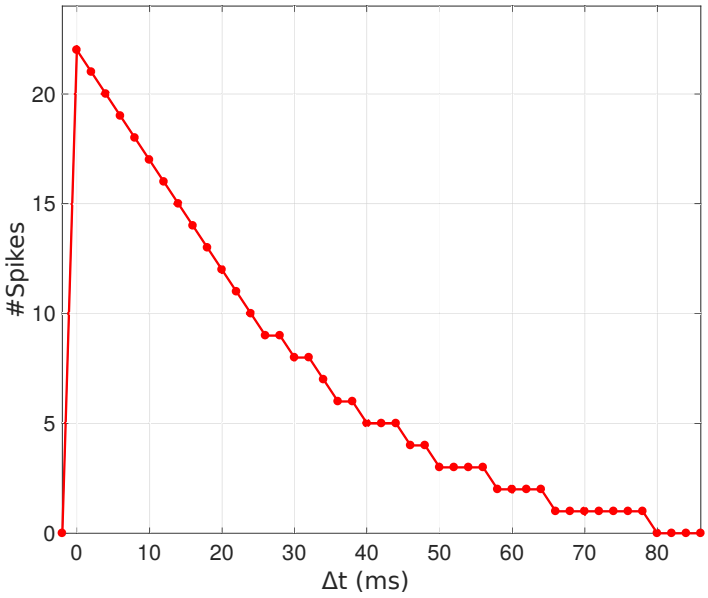


Figure 7.3: Tuning curve of the sEMD implemented on SpiNNaker. The number of spikes within a burst of activity in response to varying Δt at the two respective inputs. The implementation on SpiNNaker shows the same direction selectivity with high-dynamic range as the software and analogue hardware model.

of the sEMD. We obtained the tuning curve in the same way as described earlier (see Results). This preliminary result suggests that we can use the sEMD model on SpiNNaker. Due to the small size as well as the direct eDVS interface we can perform motion estimation and extract a collision avoidance direction in closed-loop. These experiments, however, are beyond the scope of the present paper.

7.2.3 SpiNNaker simulation parameters

Table 7.3: SpiNNaker neuron & synapse parameters

Neuron	**Time constant** [ms]	I_{offset}	
	38	0	
Synapse	**Time constant** [ms]	**Exc. syn. weight**	**Inh. syn. weight**
Facilitation	25	0.5	
Trigger	25	0.5	
General parameters			
C_m	0.25 pF		
V_r	-70.0 mV		
V_{th}	-40.0 mV		
V_{rest}	25 mV		
τ_{refrac}	0.5 ms		
$step$	0.5		

7.2.4 Brian simulation parameters

Table 7.4: Network neuron & synapse parameter. STF is short for spatio-temporal filtering. VI stands for Vertical Integrator.

Neuron	**Leak current** [pA]		
STF	22		
sEMD	18		
VI	18		
WTA	18		
Inhibitor	18		
	Frequency [Hz]	**Weight** [pA]	**Time constant**[ms]
Bias	17	25	75
Synapse	**Time constant** [ms]	**Exc. syn. weight**	**Inh. syn. weight**
DVS to STF	55	27	
STF to sEMD	35	49	
sEMD to VI	25	1.3	
VI to WTA	500		0.4
WTA recurrent	17	27	
WTA to Inhibitor	15	30	
Inhibitor to WTA	220		140
General parameters			
C_m	1 pF		
V_r	0 V		
$V_t h$	0.8 V		
U_t	25 mV		

7.2.5 Bias value of the spiking elementary motion detector chip

Table 7.5: Chip biases. Note that this is a test chip thus many biases are set to 0 in order to turn off the respective blocks which were not used in this study. Fine value ranges from 0 to 255

Bias	Coarse [A]	Fine	Bias	Coarse [A]	Fine	Bias	Coarse [A]	Fine
SY_{PWLK_P}	3.2u	53	$std\|tau\_p$	105p	48	$ca*\|vpot\_n$	24u	255
$neu_{mon}\|IF_{BUF\_P}$	0.4u	100	$std\|vth_{dpi\_p}$	3.2u	150	$emd\|wei\_n$	24u	212
$neu\|IF_{DC\_P}$	15p	0	$std\|tau_{dpi\_p}$	15p	0	$bpf\|wei_{std\_n}$	15p	0
$neu\|IF_{AHW\_P}$	15p	0	$exc\|vth_{dpi\_p}$	15p	100	$emd\|vw_{dpi\_0\_n}$	24u	208
$ca\|vpw\_p$	15p	0	$exc\|tau_{dpi\_p}$	24u	255	$emd\|vw_{dpi\_1\_n}$	15p	1
$ca\|vtau_{ca\_p}$	15p	0	$inh\|vth_{dpi\_p}$	15p	0	$cacmp\|vcmp\_p$	15p	0
$ca\|vtau_{dpi\_p}$	15p	0	$inh\|tau_{dpi\_p}$	15p	0	$bpf\|wei\_std$	15p	0
$ca\|vth_{d\_p}$	15p	0	PA_BUF_P	15p	0	$stf\|vlim\_n$	15p	0
$ca\|vth_{dpi\_p}$	15p	0	$neu\|thr\_n$	15p	100	$stf\|tau\_n$	15p	0
$ca\|vth_{p\_p}$	15p	0	$neu\|tau\_n$	820p	35	std_cacmp	0.4u	100
$ca\|vb_{bs\_p}$	15p	0	$neu\|rfr\_n$	6.5n	34	PA_BUF_N	24u	50
$ca\|vth_{w\_p}$	15p	0	$neu\|nmda\_n$	15p	0	$exc\|vw_{dpi\_n}$	24u	200
$ca\|pwlk\_p$	15p	0	$neu\|casc\_n$	15p	0	$inh\|vw_{dpi\_n}$	15p	0
$ca\|pa_{lload\_p}$	15p	0	$neu\|ahthr\_n$	15p	0	$C2F_{SWITCH\_N}$	3.2u	79
$ca\|pa_{woffst\_p}$	15p	0	$neu\|ahtau\_n$	15p	0	$C2F_{PWLK\_P}$	6.5n	80
$bpf\|wei_{stp\_p}$	15p	0	$ca\|woffst\_n$	15p	0	$C2F_{TEST\_P}$	15p	0
$bpf\|tau_{std\_p}$	15p	0	$ca*\|vw_{ini\_n}$	15p	0	$C2F_{REFL\_N}$	15p	0
$bpf\|vb_{ota\_p}$	15p	0	$ca*\|vc_{post\_n}$	15p	0	BUFFER	0.4u	255
$bpf\|tau_{dpi\_p}$	15p	0	$ca*\|vc_{pre\_n}$	15p	0	SSP	6.5n	7
$cacmp\|vw2_{dpi\_p}$	15p	0	$ca*\|vdel\_n$	24u	255	SSN	24u	15
$stf\|tau_{dpi\_p}$	105p	50	$ca*\|vdep\_n$	24u	255			

7.3 Online Clustering of Temporal Activity

For the verification experiment of $L_{2/3}$ running on DYNAP (see Section 4.4.3) we use the DYNAP board #27. We only used *Core 0* and *Core 1* of *Chip 0*.
All other cores were turned off by setting the IF_{TAU1_N} bias to coarse value of 7 and the fine value to 100 and by setting the IF_{DC_P} to coarse value of 0 and the fine value to 1.
The code reproduce the hardware experiment is online[1]. The code to reproduce the ssoftware results is also online[2]. Below we rport the bias setting for the two cores we used to emulate $L_{2/3}$. *Core 0* emulates the excitatory population (see Table 7.6), whereas *Core 1* emulates the inhibitory interneurons (see Table 7.7).

Table 7.6: Bias setting for Core 0 of Chip 0 of DYNAP board #27.

Bias name Coarse	**Fine**	**Bias name**	**Coarse**	**Fine**	
IF_{AHTAU_N}	0	1	$NPDPIE_{TAU_{F_P}}$	1	180
IF_{AHTHR_N}	0	1	$NPDPIE_{TAU_{S_P}}$	1	90
IF_{AHW_P}	0	1	$NPDPIE_{THR_{F_P}}$	2	60
IF_{BUF_P}	4	80	$NPDPIE_{THR_{S_P}}$	2	60
IF_{CASC_N}	0	1	$NPDPII_{TAU_{F_P}}$	1	70
IF_{DC_P}	0	1	$NPDPII_{TAU_{S_P}}$	7	255
IF_{NMDA_N}	0	1	$NPDPII_{THR_{F_P}}$	3	64
IF_{RFR_N}	3	100	$NPDPII_{THR_{S_P}}$	0	10
IF_{TAU1_N}	5	180	$PS_{WEIGHT_{EXC_{F_N}}}$	7	80
IF_{TAU2_N}	7	255	$PS_{WEIGHT_{EXC_{S_N}}}$	6	110
IF_{THR_N}	1	39	$PS_{WEIGHT_{INH_{F_N}}}$	5	160
$PULSE_{PWLK_P}$	4	11	$PS_{WEIGHT_{INH_{S_N}}}$	0	10

[1] https://code.ini.uzh.ch/mmilde/OCTA/blob/dev/octa/notebooks/dynap_octa.ipynb
[2] https://code.ini.uzh.ch/mmilde/OCTA

Table 7.7: Bias setting for Core 1 of Chip 0 of DYNAP board #27.

Bias name	**Coarse**	**Fine**	**Bias name**	**Coarse**	**Fine**
IF_{AHTAU_N}	0	1	$NPDPIE_{TAU_{F_P}}$	1	110
IF_{AHTHR_N}	0	1	$NPDPIE_{TAU_{S_P}}$	1	110
IF_{AHW_P}	0	1	$NPDPIE_{THR_{F_P}}$	3	231
IF_{BUF_P}	4	80	$NPDPIE_{THR_{S_P}}$	3	230
IF_{CASC_N}	0	1	$NPDPII_{TAU_{F_P}}$	1	110
IF_{DC_P}	0	1	$NPDPII_{TAU_{S_P}}$	7	255
IF_{NMDA_N}	0	1	$NPDPII_{THR_{F_P}}$	3	230
IF_{RFR_N}	3	100	$NPDPII_{THR_{S_P}}$	0	10
IF_{TAU1_N}	5	180	$PS_{WEIGHT_{EXC_{F_N}}}$	1	10
IF_{TAU2_N}	7	255	$PS_{WEIGHT_{EXC_{S_N}}}$	7	85
IF_{THR_N}	1	37	$PS_{WEIGHT_{INH_{F_N}}}$	5	50
$PULSE_{PWLK_P}$	4	11	$PS_{WEIGHT_{INH_{S_N}}}$	1	10